

**FATIGUE STRENGTH REDUCTION MODEL:  
RANDOM3 and RANDOM4 USER MANUAL**

11-27  
1917-12  
808.

Prepared by :

Lola Boyce, Ph.D., P.E.  
Thomas B. Lovelace

**APPENDIX 2  
of Annual Report  
of Project Entitled  
Development of Advanced Methodologies  
for Probabilistic Constitutive Relationships  
of Material Strength Models**

NASA Grant No. NAG 3-867

Prepared for :

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
Lewis Research Center  
Cleveland, OH 44135**

(NASA-CR-184796-APP-2) FATIGUE STRENGTH  
REDUCTION MODEL: RANDOM3 AND RANDOM4 USER  
MANUAL. APPENDIX 2: DEVELOPMENT OF ADVANCED  
METHODOLOGIES FOR PROBABILISTIC CONSTITUTIVE  
RELATIONSHIPS OF MATERIAL STRENGTH MODELS

N89-19582

Unclas  
G3/39 0191792

**The Division of Engineering  
The University of Texas at San Antonio  
San Antonio, TX 78285  
January, 1989**

**FATIGUE STRENGTH REDUCTION MODEL:  
RANDOM3 and RANDOM4 USER MANUAL**

**Prepared by :**

**Lola Boyce, Ph.D., P.E.  
Thomas B. Lovelace**

**APPENDIX 2  
of Annual Report  
of Project Entitled  
Development of Advanced Methodologies  
for Probabilistic Constitutive Relationships  
of Material Strength Models**

**NASA Grant No. NAG 3-867**

**Prepared for :**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
Lewis Research Center  
Cleveland, OH 44135**

**The Division of Engineering  
The University of Texas at San Antonio  
San Antonio, TX 78285  
January, 1989**

## TABLE OF CONTENTS

SECTION	PAGE
1.0 Introduction . . . . .	1
2.0 Theoretical Background . . . . .	2
3.0 Input Data . . . . .	4
4.0 Sample Problem . . . . .	7
5.0 References . . . . .	11
6.0 Appendix A: Physical Quantities, Symbols, and Units . . . . .	12
7.0 Appendix B: RANDOM3 Sample Problem: Source, Input and Output Files . .	13
8.0 Appendix C: RANDOM4 Sample Problem Source, Input and Output Files . .	37
9.0 Appendix D: IMSL Subroutine Calls from RANDOM3 and RANDOM4 . . . .	76
10.0 Appendix E: SAS/GRAPH Program for RANDOM3 and RANDOM4 . . . . .	77

## 1.0 INTRODUCTION

This User Manual documents the FORTRAN programs RANDOM3 and RANDOM4. They are based on fatigue strength reduction, using a probabilistic constitutive model. They predict the random lifetime of an engine component to reach a given fatigue strength (see Section 2.0, Theoretical Background).

Included in this Manual are details regarding the theoretical backgrounds of RANDOM3 and RANDOM4, input data instructions and sample problems illustrating the use of RANDOM3 and RANDOM4. Appendix A gives information on the physical quantities, their symbols, FORTRAN names and both SI and U.S. Customary units. Appendix B and C include photocopies of the actual computer printout corresponding to the sample problems. Appendices D and E detail the IMSL, Version 10<sup>1</sup>, subroutines and functions called by RANDOM3 and RANDOM4 and SAS/GRAPH<sup>2</sup> programs that can be used to plot both the probability density functions (p.d.f.) and the cumulative distribution functions (c.d.f.).

## 2.0 THEORETICAL BACKGROUND

Fatigue strength data are usually presented as cycles to failure for each of several stress amplitudes, the familiar S-N diagram. Results indicate that for lower stress amplitudes the cycles (or time) to failure increases. Thus, a power curve fit through the data yields a monotonically decreasing curve. In general, this curve is represented as

$$S = [N/C']^{-1/m'} \quad (6)$$

where the primitive variables in this equation are as follows: S is the applied constant amplitude alternating stress at failure or fatigue strength, N is number of cycles, C' is a material parameter that varies from specimen to specimen and m' is a material constant.<sup>3</sup> Equation (6) can be written in terms of "cycles to reach a given fatigue strength" as

$$N = C'S^{-m'} \quad (7)$$

Recently another fatigue strength reduction model has been proposed that takes into account the effect of temperature as well as other parameters that affect strength.<sup>4</sup> The general form of the constitutive relationships for this model is applied to the constituents of high temperature composite materials. Specifically, it is applied herein for the case of a single material constituent. The mechanical property of interest is fatigue strength which is expressed in terms of primitive variables, including the general categories of temperature, mechanical cycles and mean stress. For these categories, the relationship becomes

$$\frac{S}{S_o} = \left[ \frac{T_F - T}{T_F - T_o} \right]^n \left[ \frac{S_F - \sigma}{S_F - \sigma_o} \right]^m \left[ \frac{\log N_{MF} - \log N_M}{\log N_{MF} - \log N_{MO}} \right]^q \quad (8)$$

where S is the applied constant amplitude alternating stress at failure (fatigue strength) at current (or operating) temperature, T, mean stress,  $\sigma$ , and mechanical cycle,  $N_M$ .  $S_o$  is fatigue strength at reference temperature,  $T_o$  (usually room temperature), reference mean stress (or residual stress),  $\sigma_o$ , and reference mechanical cycle,  $N_{MO}$ . Also,  $T_F$  is the final or melting temperature of the material,  $S_F$  is the final or tensile strength of the material, and  $N_{MF}$  is the final mechanical cycle or lifetime. Empirical parameters, n, m, and q, are determined from available experimental data or estimated from anticipated behavior of the particular product term.<sup>5</sup> Note that the term containing mechanical cycles is expressed in terms of the log of cycles rather than cycles. This formulation is attractive when  $N_M$  and  $N_{MO}$  are small compared to  $N_{MF}$ . The equation may be solved for  $N_M$ , or the "cycles to reach a given fatigue strength." The expression is

$$N = 10 \exp \left[ \log N_{MF} - \left( \log N_{MF} - \log N_{MO} \right) \left[ \frac{S}{S_o \left[ \frac{T_F - T}{T_F - T_o} \right]^n \left[ \frac{S_F - \sigma}{S_F - \sigma_o} \right]^m} \right]^{1/q} \right] \quad (9)$$

For values typical of a cast nickel base-superalloy subjected to typical loads and temperatures, equation (9) indicates increasing life for decreasing temperature, decreasing tensile mean stress, and decreasing applied alternating stress. It indicates decreasing life for increasing temperature, decreasing compressive mean stress, and increasing applied alternating stress. Therefore, equation (9) predicts observed trends in general.

Probabilistic analysis, via simulation, yields the distribution of the dependent random variable, cycles,  $N$ . A probability density function (p.d.f.) of cycles is generated using the maximum penalized likelihood method for RANDOM3. For RANDOM4, a p.d.f. of cycles is generated using the maximum entropy method. Maximum entropy uses Jaynes' principle which says that "the minimally prejudiced distribution is that which maximizes the entropy subjected to the constraints supplied by the given information."<sup>6</sup>

### 3.0 INPUT DATA

Data input for RANDOM3 and RANDOM4 is user friendly and easy to manipulate (see, for example, the file entitled NORMAL.INP, in Section 4.0). The first twelve lines of input have the same format, 2E12.4 and the last two lines differ. The last two lines of input have the formats I3,2X,I3,2X,2E12.4,2X,I3 and I3, respectively. A brief, line by line description is given along with an example for each line (NOTE: the ruler is to aid the user in formatting and is not a part of the input). A table listing the physical quantities, their units and symbols is given in Appendix A.

- ### 1. Random Number Generator Seed, ISEED, and Sample Size, NTOT

**EXAMPLE:**

123456789012345678901234567890

- ## 2. Ultimate Tensile Strength, SF

**EXAMPLE:**

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
										900.0000										45.0000									

- ### 3. Log of Final Cycle, NMF

**EXAMPLE:**

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
										8.0000										0.8000									

- #### 4. Reference Fatigue Strength, SO

**EXAMPLE:**

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
										500.0000										25.0000									

- ## 5. Log of Reference Cycle, NMO

**EXAMPLE:**

123456789012345678901234567890
7.0000 0.7000

6. Current Fatigue Strength, S

EXAMPLE:

123456789012345678901234567890  
250.0000 12.0000

7. Residual Compressive Stress, SIGO

EXAMPLE:

123456789012345678901234567890  
20.0000 1.0000

8. Current Mean Stress, SIG

EXAMPLE:

123456789012345678901234567890  
150.0000 7.5000

9. Temperature Exponent, XXN, Stress Exponent, XXM, and Cycle Exponent, XXQ

EXAMPLE:

123456789012345678901234567890  
0.5000 0.0150

10. Melting Temperature, TF

EXAMPLE:

123456789012345678901234567890  
1500.0000 75.0000

11. Reference Temperature, TO

EXAMPLE:

123456789012345678901234567890  
20.0000 0.6000



12. Current Temperature, T

EXAMPLE:

<u>123456789012345678901234567890</u>									
850.0000					25.0000				

13. The DESPL<sup>1</sup> parameters are NODE, INIT, ALPHA, EPS, and MAXIT and are entered in that order as follows:

EXAMPLE:

<u>123456789012345678901234567890</u>										
21		0		20.0000			1.0E-05		30	

14. The DESPL parameter, IOPT, is entered as follows:

EXAMPLE:

<u>1234567890</u>									
2									

#### 4.0 SAMPLE PROBLEMS FOR RANDOM3 AND RANDOM4

The objective of these programs is to predict the random lifetime to reach a given fatigue strength for an engine component. The theory is based on fatigue strength reduction, using a probabilistic constitutive model. The only difference between RANDOM3 and RANDOM4 is the method used to generate p.d.f. estimates. RANDOM3 uses maximum penalized likelihood, while RANDOM4 uses maximum entropy (see Section 2.0, Theoretical Background). RANDOM3 and RANDOM4 input parameters are given in Table A2.1.

TABLE A2.1 RANDOM3 and RANDOM4 input (SI units)

FORTRAN Name	Distribution Type	Mean	Standard Deviation	
			(Value)	(% of Mean)
SF	normal	900.0	45.0	(3%)
NMF	lognormal	8.0	0.8	(10%)
SO	lognormal	500.0	25.0	(5%)
NMO	lognormal	7.0	0.7	(10%)
S	lognormal	250.0	12.5	(5%)
SIGO	lognormal	-20.0	-1.0	(1%)
SIG	lognormal	150.0	7.5	(5%)
XXN	normal	0.5	0.015	(0.3%)
XXM	normal	0.5	0.015	(0.3%)
XXQ	normal	0.5	0.015	(0.3%)
TF	normal	1500.0	45.0	(3%)
TO	normal	20.0	0.6	(3%)
T	normal	850.0	25.5	(3%)

The input is entered in the following format in a file entitled NORMAL.INP.

```

1234567890123456789012345678901234567890
  1                                40
900.0000      45.0000
  8.0000      0.8000
500.0000      25.0000
  7.0000      0.7000
250.0000      12.5000
20.0000       1.0000
150.0000      7.5000
  0.5000      0.0150
1500.0000     75.0000
20.0000       0.6000
850.0000     25.5000
21  0          20.00      1.0E-05  30
  2

```

Execution of RANDOM3 and RANDOM4 (source code entitled NR3.FOR and NR4.FOR, respectively) produces files entitled RANDM33 and RANDM44. These give intermediate results (see Appendices B and C). Execution also produces plotfiles entitled PLOT1 and PLOT2 (see Appendices B and C). These files are used to plot the X and Y axes of the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.), respectively, generated by RANDOM3 and RANDOM4. The plots are drawn from the plotfiles by the SAS/GRAPH graphing program (see Appendix D). These plots for the sample problem are shown Figures 1, 2, 3, and 4. This same sample problem has been reported in Boyce and Chamis.<sup>7</sup> There, however, it utilized U.S. Customary units and older versions of RANDOM3 and RANDOM4 (using IMSL Version 9.2 subroutines).

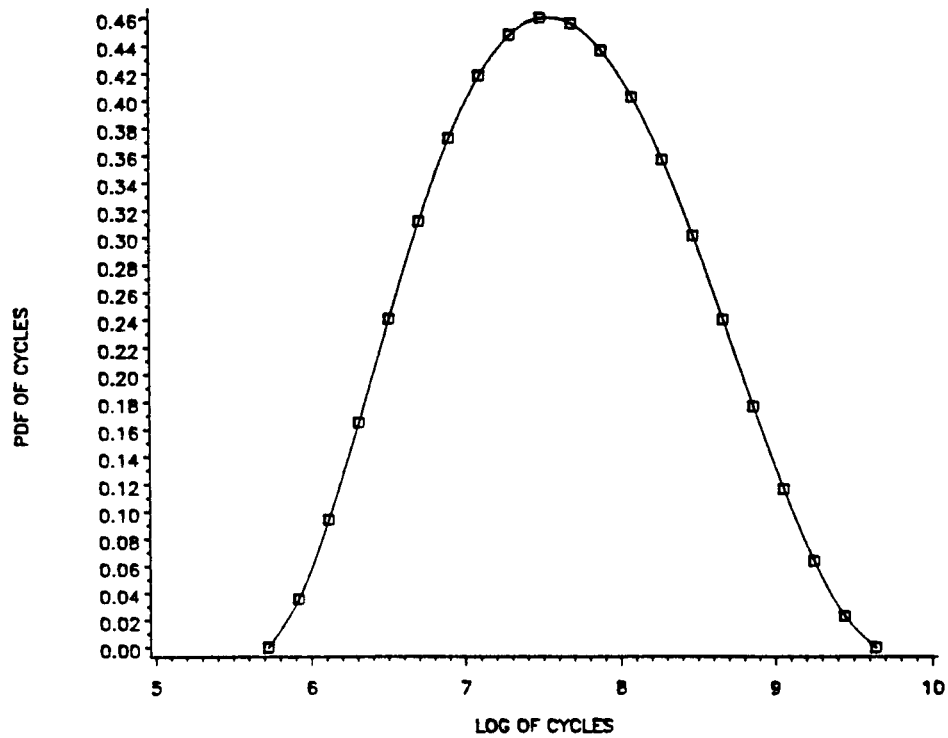


Fig. A2.1 p.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum penalized likelihood method of p.d.f. generation.

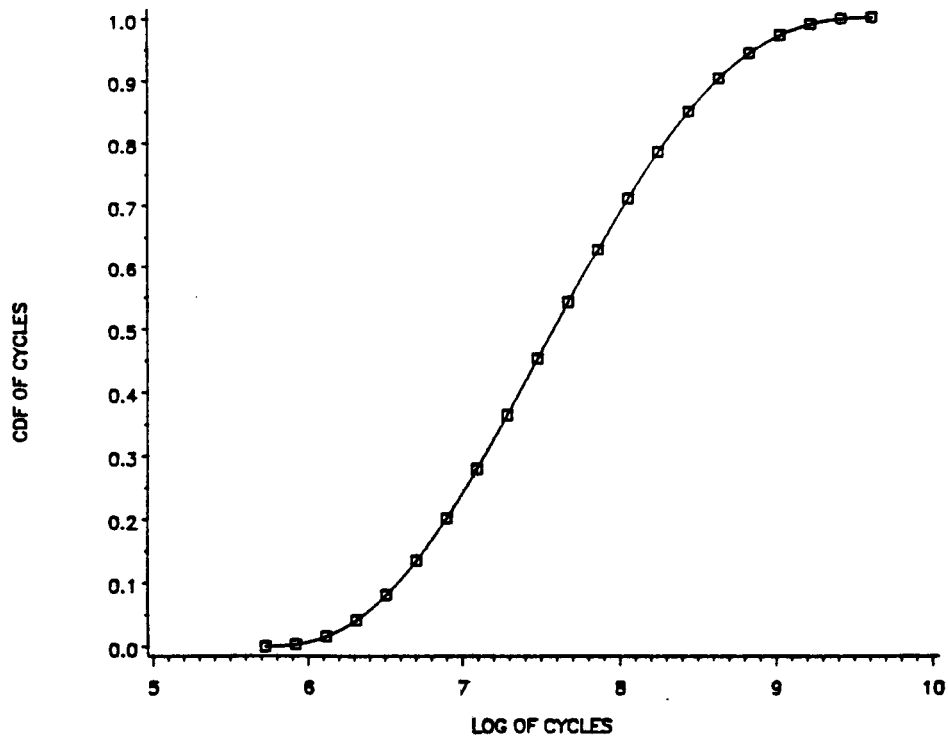


Fig. A2.2 c.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum penalized likelihood method of p.d.f. generation.

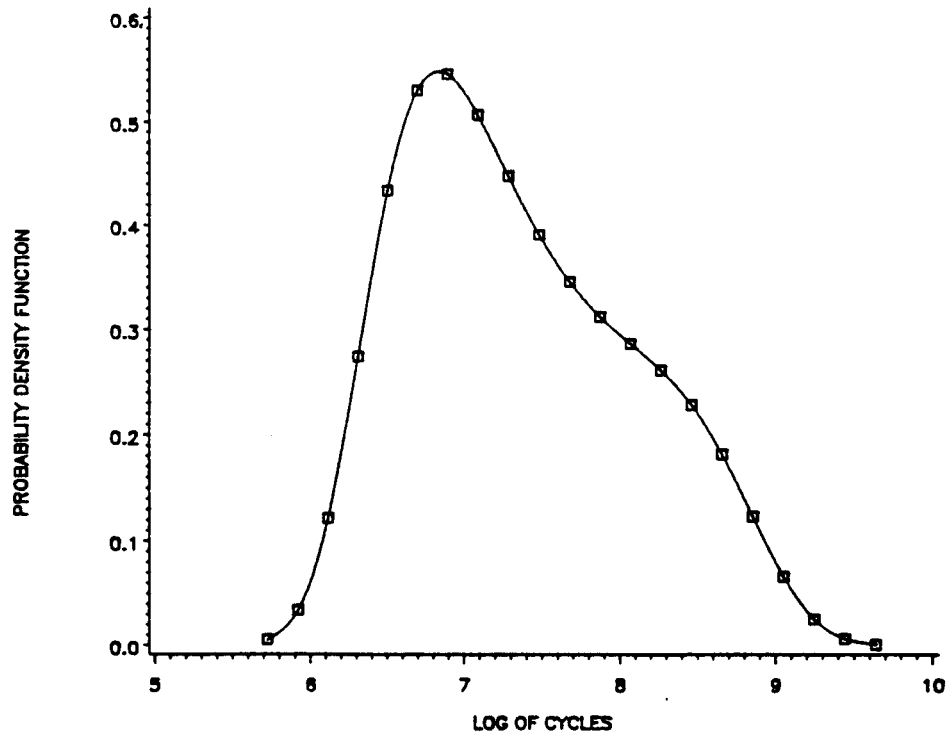


Fig. A2.3 p.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum entropy method of p.d.f. generation.

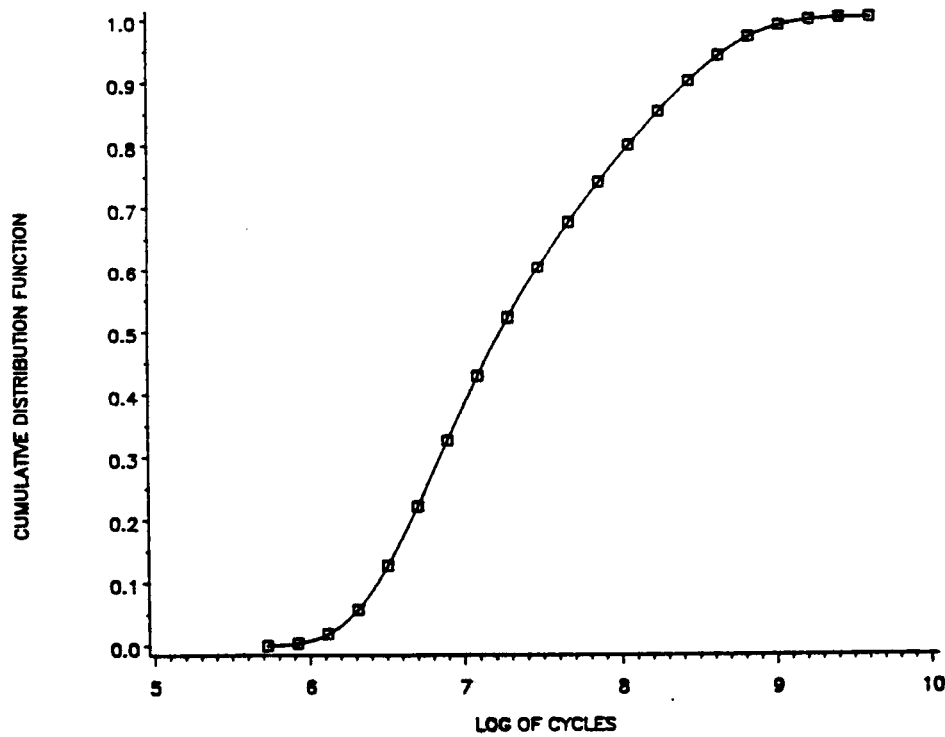


Fig. A2.4 c.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum entropy method of p.d.f. generation.

## 5.0 REFERENCES

- <sup>1</sup> IMSL, "STAT/LIBRARY, FORTRAN Subroutines for Statistical Analysis", Houston, Texas
- <sup>2</sup> SAS Institute, Inc., SAS/GRAPH User's Guide, Version 5 Edition, Cary NC: SAS Institute, Inc., 1985, 596 pp.
- <sup>3</sup> Madsen, H.O., "Bayesian Fatigue Life Prediction," Probabilistic Methods in the Mechanics of Solids and Structures, S. Eddwertz and N.C. Lind, Eds., Proceedings of the IUTAM Symposium, Stockholm, Sweden, 1984, pp. 395-406.
- <sup>4</sup> Hopkins, D.A. and Chamis, C.C., "A Unique Set of Micromechanics Equations for High Temperature Metal Matrix Composites," NASA TM87154, Nov., 1985.
- <sup>5</sup> Chamis, C.C. and Hopkins, D.A., "Thermoviscoplastic Nonlinear Constitutive Relationships for Structural Analysis of High Temperature Metal Matrix Composites," NASA TM 87291, Nov., 1985.
- <sup>6</sup> Siddall, J.N., "A Comparison of Several Methods of Probabilistic Modeling," Proceedings of the Computers in Engineering Conference, ASME, San Diego, CA, Vol. 4, 1982, pp. 231-238.
- <sup>7</sup> Boyce, L. and Chamis, C.C., "Probabilistic Constitutive Relations for Cyclic Material Strength Models," Proceedings, 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, VA, 1988.

## 6.0 APPENDIX A

### PHYSICAL QUANTITIES, SYMBOLS, AND UNITS

The physical quantities, their symbols and units for the fatigue crack growth model are given in the following table.

Table A2.2 Physical quantities, symbols, and units for fatigue crack growth model for RANDOM3 and RANDOM4.

Physical Quantity	Theory Symbol	FORTRAN Name	Units	
			SI	U.S.
Ultimate Tensile Strength	SF	SF	MPa	ksi
Final Cycle (lifetime)	$N_{MF}$	NMF	dimensionless	
Reference Fatigue Strength	SO	SO	MPa	ksi
Reference Cycles	$N_{MO}$	NMO	dimensionless	
Current Fatigue Strengths	S	S	MPa	ksi
Residual Compressive Stress	$\sigma_o$	SIGO	MPa	ksi
Current Mean Stress	$\sigma$	SIG	MPa	ksi
Empirical Material Parameters	n	XXN	dimensionless	
	m	XXM	dimensionless	
	q	XXQ	dimensionless	
Melting Temperature	TF	TF	°C	°F
Reference Temperature	TO	TO	°C	°F
Current Temperature	T	T	°C	°F

## 7.0 APPENDIX B

### RANDOM3 SAMPLE PROBLEM: SOURCE, INPUT AND OUTPUT FILES





```

JOB=COMP3-USUSQ530-RT=80-T=30-HF=300000Y.
ACCOUNT=UW=LULAB.
DELETE,PUN=NR3BLD,ID=SHBOYCE.
DET=7,LIST.
REWIND,DN=1BLD.
BUILD,I=0,OEL=0.
SAVE,DN=$NBL,CN=NR3BLD,ID=SHBOYCE,ED=-1.
DELETE,PDN=NR3BLD,ID=SHBOYCE,ED=-1.
/EOF
***** C CHANNELS MICROMECHANICS CONSTITUTIVE EQUATIONS;
***** DIMENSIONALIZED AND APPLIED TO FATIGUE STRENGTH
***** INTEGER NTOT, ISEED, M, INIT, NMISSE, MAXIT, NODE
***** REAL YM, XS, YM, YS, EPS, P, RWKSF, S12, ALPHA
COMMON /WORKSP/ RWKSP
DIMENSION SF(1000),XLNMF(10000),SD(10000)
DIMENSION XLNM(1000),SIG(1000)
DIMENSION SIGO(1000),SIGI(1000)
DIMENSION XNM(1000),XXM(1000),XQ(1000)
DIMENSION XNH(1000),XNDXS(1000)
DIMENSION IF(1000),TD(1000),TENS(1000),OBTXT(1000)
DIMENSION STAT(10000),GENS(1000),B8(999),FF(999)
DIMENSION ENDS(1000),FP(999)
DIMENSION XXXX(999),PPPP(999)
DIMENSION BRRR(999),FFFF(999)
DIMENSION Z(1000)
EXTERNAL RNENL, RNSSET, RNNOR, DESPL, IWKN
EXTERNAL UTIME, TENSE, STRENTH, SE
C LOGNO
1001 FORMAT(5E12.4)
1002 FORMAT(I12,I12)
1003 FORMAT(I4,I4)
1004 FORMAT(I4)
1005 FORMAT(I3,2X,I3,2X,2E12.4,2X,I3)
1006 FORMAT(I3)
1007 FORMAT(2E12.4)
1008 FORMAT('LOGNORMAL SF')
1009 WRITE(3,1002) ISEED,NTOT
1010 READ(3,1011) XM,XS
1011 WRITE(4,1011) XM,XS
YS = SQRT(LOG(1.0+(XS/XH)**2))
YM = LOG(XM) - 0.5*YS**2.
CALL RNSSET(ISEED)
CALL RNENL(NTOT,YM,YS,SF)
WRITE(6,2020)
2020 FORMAT('LOGNORMAL SF')
WRITE(6,1001) (SF(I),I=1,NTOT)
C LOGNO
1009 WRITE(6,1002) OF FINAL CYCLE XLNMF
WRITE(6,1002) ISEED,NTOT
READ(3,1011) XM,XS
WRITE(6,1011) XM,XS
YS = SORT( LOG(1.0+(XS/XH)**2) )
YM = LOG(YM) - 0.5*YS**2
CALL RNSSET(ISEED)
CALL RNENL(NTOT,YM,YS,XLNMF)
WRITE(6,2021)
2021 FORMAT('LOGNORMAL XLNMF')
WRITE(6,1001) (XLNMF(I),I=1,NTOT)
C LOGNO
1009 WRITE(6,1002) OF FINAL CYCLE STRENGTH AT REFERENCE CONDITIONS, SD
WRITE(6,1002) ISEED,NTOT
READ(3,1011) XM,XS
WRITE(6,1011) XM,XS
YS = SORT( LOG(1.0+(XS/XH)**2) )
YM = LOG(YM) - 0.5*YS**2
CALL RNSSET(ISEED)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

C 2022 WRITE(6,2022) LOGNORMAL SIGO
C 2022 WRITE(6,1001) (SIG(I),I=1,NTOT)
C LOGNORMAL LOG OF REFERENCE CYCLES, XLNMO
READ(3,1002) ISEED,NTOT
WRITE(6,1011) XM, XS
WRITE(6,1011) XM, XS
YS = SQR(LG(1.0+(XS/XM)**2))
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNLN(NTOT,YM,YS,XLNMO)
C 2023 WRITE(6,2023)
C 2023 WRITE(6,1001) (XLNMO(I),I=1,NTOT)
C LOGNORMAL FATIGUE STRENGTH AT CURRENT CONDITIONS, S
WRITE(6,1002) ISEED,NTOT
READ(3,1011) XM, XS
WRITE(6,1011) XM, XS
YS = SQR(LG(1.0+(XS/XM)**2))
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNLN(NTOT,YM,YS,S)
C 2024 WRITE(6,2024)
C 2024 WRITE(6,1001) (S(I),I=1,NTOT)
C DEFINE RANDOM STRESSES
C LOGNORMAL REFERENCE STRESS, SIGO
WRITE(6,1002) ISEED,NTOT
READ(3,1011) XM, XS
WRITE(6,1011) XM, XS
YS = SQR(LG(1.0+(XS/XM)**2))
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNLN(NTOT,YM,YS,SIGO)
C CHANGE SIGO TO NEGATIVE VALUES FOR COMPRESSIVE
DO 201 I = 1,NTOT
SIGO(I) = -SIGO(I)
201 CONTINUE
C 2036 WRITE(6,2036)
C 2036 WRITE(6,1001) (SIGO(I),I=1,NTOT)
C LOGNORMAL CURRENT STRESS, SIG
WRITE(6,1002) ISEED,NTOT
READ(3,1011) XM, XS
WRITE(6,1011) XM, XS
YS = SQR(LG(1.0+(XS/XM)**2))
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNLN(NTOT,YM,YS,SIG)
C 2037 WRITE(6,2037)
C 2037 WRITE(6,1001) (SIG(I),I=1,NTOT)
C NORMAL EXPONENTS, XXN,XXH,XXQ
WRITE(6,1002) ISEED,NTOT
READ(3,1011) YM, YS
WRITE(6,1011) YM, YS
CALL RNSET(ISEED)
CALL RNLN(NTOT,XXN)
DO 202 I=1,NTOT
XXN(I) = YS**XXN(I)+YM
202 CONTINUE
202 WRITE(6,2025)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
2025 FORMAT(1, NORMAL, XNM)
WRITE(6,1001)XNM(I),I=1,NTOT)
CALL RNSET(ISEED)
CALL RNNOR(NTOT,XNM)
DO 203 I=1,NTOT
  XNM(I)=YS*XXM(I)+YM
203 CONTINUE
204 WRITE(6,2026)
FORMAT(1, NORMAL, XNM)
WRITE(6,1001)XNM(I),I=1,NTOT)
CALL RNSET(ISEED)
CALL RNNOR(NTOT,XXQ)
DO 204 I=1,NTOT
  XXQ(I)=YS*XXQ(I)+YM
204 CONTINUE
205 WRITE(6,2027)
FORMAT(1, NORMAL, XXQ)
WRITE(6,1001)XXQ(I),I=1,NTOT)
C NORMAL TEMPERATURES, TF, TO, T
C NORMAL FINAL (MELTING) TEMPERATURE, TF
WRITE(6,1002)ISEED,NTOT
READ(3,1011) YM, YS
WRITE(6,1011) YM, YS
CALL RNSET(ISEED)
CALL RNNOR(NTOT,IF)
DO 205 I=1,NTOT
  IF(I)=YS*IF(I)+YM
205 CONTINUE
206 WRITE(6,2046)
FORMAT(1, NORMAL, IF)
WRITE(6,1001)IF(I),I=1,NTOT)
C NORMAL REFERENCE TEMPERATURE, TO
WRITE(6,1002)ISEED,NTOT
READ(3,1011) YM, YS
WRITE(6,1011) YM, YS
CALL RNSET(ISEED)
CALL RNNOR(NTOT,IO)
DO 206 I=1,NTOT
  IO(I)=YS*IO(I)+YM
206 CONTINUE
207 WRITE(6,2047)
FORMAT(1, NORMAL, TO)
WRITE(6,1001)TO(I),I=1,NTOT)
C NORMAL CURRENT TEMPERATURE, T
WRITE(6,1002)ISEED,NTOT
READ(3,1011) YM, YS
WRITE(6,1011) YM, YS
CALL RNSET(ISEED)
CALL RNNOR(NTOT,I)
DO 207 I=1,NTOT
  T(I)=YS*T(I)+YM
207 CONTINUE
208 WRITE(6,2048)
FORMAT(1, NORMAL, I)
WRITE(6,1001)T(I),I=1,NTOT)
C CALCULATE LOG OF CURRENT CYCLES, LOG XNM
DO 102 I=1,NTOT
  RS=((SF(I)-SIG(I))/(SF(I)-SIGO(I)))*XXM(I)
  TEMP=((TF(I)-T(I))/(TF(I)-TO(I)))*XXN(I)
  XNM1=(S(I)/(SO(I)*TEMP*RS))*XXQ(I)
  XNM2=(XLNME(I)-(XLNME(I)-XLNME(I))*XXNH(I))
  IF (XNM2.LT.0.0) XNM2=0.0
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
XNM1)-XNM2
102 CONTINUE
WRITE(6,2028)
2028 FORMAT(' LOG OF CYCLES TO REACH MEAN FATIGUE STR = ',,
1,250 MPA')
-- C SORT LOG OF CYCLES
CALL SORT(XNM,NTOT)
WRITE(6,2029)
2029 FORMAT(' SORTED LOG OF CYCLES')
C CALCULATE PDF OF LOG OF CURRENT CYCLES, LOG XNM
READ(3,1009)NODE,INIT,ALPHA,EPS,MAXIT
WRITE(6,985)
985 FORMAT(' DESPL PARAMETERS')
WRITE(6,1009)NODE,INIT,ALPHA,EPS,MAXIT
BND1(1)=XNM(1)-0.05*XNM(1)
BND2(2)=XNM(NTOT)+0.05*XNM(NTOT)
WRITE(6,986)BND1(1),BND2(2)
986 FORMAT(' BND1(1),BND2(2)=',E12.4,1X,E12.4)
CALL DESPL(NTOT,XNM,NODE,BND1,INIT,ALPHA,MAXIT,EPS,DENS,STAT,
1NMIS)
WRITE(6,980)
980 FORMAT(' PDF OF LOG OF CURRENT CYCLES, LOG XNM, Y AXIS OF PDF PLOT')
WRITE(6,1001)(DENS(I),I=1,NODE)
WRITE(6,981)
981 FORMAT(' OUTPUT STATISTICS')
WRITE(6,1001)(STAT(I),I=1,4)
982 FORMAT(' NUMBER OF MISSING VALUES')
WRITE(6,1010)NMIS
C CALCULATE WINDOW WIDTH, HH
HH=(BND2(2)-BND1(1))/(NODE-1)
C CALCULATE VALUES OF LOG OF CURRENT CYCLES AT WHICH PDF IS ESTIMATED;
C ALSO CALLED "NODE" VALUES
DO 6001,I=1,NODE-2
BND1(I+2)=BND1(1)+(I*HH)
6001 CONTINUE
WRITE(6,983)
983 FORMAT(' LOG OF CURRENT CYCLES, LOG XNM')
WRITE(6,1001)(BND1(I),I=1,NODE)
C REORDER BND1 FOR PLOTTING
SAVE1 = BND1(2)
SAVE2 = BND1(NODE)
BND1(2)=BND1(3)
BND1(NODE)=BND1(1)
DO 6002,I=1,NODE-2
BND1(I+1)=BND1(I+2)
6002 CONTINUE
BND1(NODE-1)=SAVE2
BND1(NODE)=SAVE1
WRITE(6,984)
984 FORMAT(' ORDERED LOG OF CURRENT CYCLES, LOG XNM,
1, Y AXIS, PDF, CDF PLOT')
C WRITE LOG OF CURRENT CYCLES AND PDF OF LOG OF CURRENT CYCLES,
C LOG XNM TO PLOT FILES
WRITE(14,990)
990 FORMAT(' (E12.4,1X,E12.4)')
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

991 WRITE(3,991) (BNDX(J), J=1, N)
991 FORMAT(12,4,1X,E12,4)
C CALCULATE CDF OF LOG OF CURRENT CYCLES
C
      READ(3,1010) IOPT
      WRITE(6,992)
      992 FORMAT(' GCD OF PARAMETERS ')
      WRITE(6,1010) IOPT
      X0=BNDX(1)
      DO 4003 I=1,NODE
        P=GCD(X0,IOPT,NODE,BNDS,DENS)
        BNDX(I)=X0
        X0=X0+HH
        DISTX(I)=P
      4003 CONTINUE
      994 WRITE(6,994)
      994 FORMAT(' CDF OF LOG OF CURRENT CYCLES, LOG XNM, '
     1 'Y-AXIS OF PDF, CDF, CDF PLOT, '
     1 'WRITE(6,1001)(DISTX(I), I=1,NODE)
C
      WRITE(6,993)
      993 FORMAT(' ORDERED LOG OF CURRENT CYCLES, LOG XNM, '
     1 'X-AXIS OF PDF, CDF PLOT, '
     1 'WRITE(6,1001)(BNDX(I), I=1,NODE)
      WRITE(6,1001)(BNDX(I), I=1,NODE)
C WRITE LOG OF CURRENT CYCLES AND CDF OF LOG OF CURRENT
C TO THE PLOT FILES
      19
      WRITE(35,990)
      WRITE(35,991)(BNDX(J),DISTX(J),J=1,NODE)
      STOP
      END
      SUBROUTINE SORT(X,N)
      DIMENSION Y(10000)
      N1=N-1
      DO 1 I=1, N1
        J=I+1
        DO 2 K=J, N
          IF (Y(I).LT.Y(K)) GO TO 2
          TEMP=Y(I)
          Y(I)=Y(K)
          Y(K)=TEMP
        2 CONTINUE
      1 CONTINUE
      RETURN
      END
C----- INSL Name: D3SPL/D03SPL (Single/Double precision version)
C-----
C----- Computer: IBM/SINGLE
C----- Revised: November 1, 1985
C----- Purpose: Nonparametric probability density function estimation
C----- estimation by the penalized likelihood method.
C----- Usage: CALL D3SPL (NOBS, X, NODE, BNDX, INIT, ALPHA, MAXIT, EPS,
C----- DENS, STAT, HESS, LDHNESS, ILOHI, DENEST, B,
C----- IPUT1, WK2)
C----- Arguments:
C----- NOBS - Number of observations. (input)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

Vector of length NBDS containing the random sample of
responses. (Input)
- Number of mesh nodes for the discrete pdf estimate.
  (Input)
- Vector of length 2 containing the minimum and maximum
  values for X(1) in BNDS(1) and BNDS(2), respectively.
  (Input)
- Initialization option. (Input)
- Positive penalty weighting factor which controls the
  smoothness of the estimate. (Input)
- Maximum number of iterations allowed in the iterative
  procedure. (Input)
- Convergence criterion. (Input)
- Vector of length NODE containing the estimated values of
  the discrete pdf at the NODE equally spaced mesh nodes.
  (Input/output if INIT=1, Output otherwise)
- Vector of length 4 containing out statistics. (Output)
  STAT(1) and STAT(2) contain the log-likelihood and the
  log-penalty terms, respectively. STAT(3) and STAT(4)
  contain the estimated mean and variance for the
  estimated density.
- Seven by NODE-1 Hessian matrix (and its factorization).
  (Output)
LDHESS - Leading dimension of HESS exactly as specified in the
dimension statement in the calling program. (Input)
ILOHI - NODE by 2 matrix containing the indices for the risk set
at each node value. (Output)
DENEST - NODE by 3 matrix containing the gradient vector, among
other quantities. (Output)
B - Vector of length NODE containing the NODE values.
  (Output)
IPVT - Pivot vector of length NODE-2. (Output)
WK2 - Work vector of length NODE-2. (Output)

Chapter: SIAI/LIBRARY Density and Hazard Estimation
Copyright: 1985 by IMSL, Inc. All Rights Reserved.
Warranty: IMSL warrants only that IMSL testing has been applied
to this code. No other warranty, expressed or implied,
is applicable.

SUBROUTINE D3SPL (NOBS, X, NODE, BNDS, INIT, ALPHA, MAXIT, EPS,
DENES, STAT, HESS, LDHESS, ILOHI, DENEST, B,
IPVT, WK2)
INTEGER NOBS, NODE, INIT, MAXIT, LDHESS, ILOHI(NODE,X),
IPVT(X)
REAL ALPHA, EPS, X(X), BNDS(2), DENES(1), STAT(1),
HESS(LDHESS,X), DENEST(NODE,X), B(X), WK2(X)
INTEGER I, IPTR, IPTR, ITER, K, KM1, KM2, KP1, KP2, M, MOLD,
NER, NOBI
REAL BK, BKMI, BSMALL, CK, CKM1, CKM2, CKMCM1, CKP1, CKP2,
CONS, EPS1, FACTOR, FK, FKM1, FKM2, FKPI, H, H2, H3,
SUM, TEMP, WK(4)
DOUBLE PRECISION SUM1, SUM2, SUM3
SPECIFICATIONS FOR SAVE VARIABLES
INTEGER MINCR(8)
SAVE MINCR
intrinsic alog,maxi,max0,min0,mod,sort
SPECIFICATIONS FOR INTRINSICS

```

```

IMPRINSIC MAXO AMAXI MINO MOD SORT SPECIFICATIONS FOR SUBROUTINES
INTEGER MAXO, MINO, MOD ALUG, AMAXI, SORT
REAL EIMES, EIPOP, EIFSH, EISII, EISTR, SADD, SAXPY, SCOPY, SHPROD, SSCAL, DSPT, LSTR9, LPERR
EXTERNAL ISMIN, NIACD, SDOT, SNRM2, SSUM
INTEGER ISMIN, NIACD, SDOT, SNRM2, SSUM
REAL SDOT, SNRM2, SSUM

DATA MINCR/5, 9, 17, 33, 65, 129, 253, 10001/

CALL EIFSH ('DSPL ') Error checks

NER = 1
IF (NOBS LT, 1) THEN
  CALL EIMES (5, 1, 'After removing all missing (NaN, not a number), values from X there are no valid observations. At least one valid observation is necessary.')
ELSE IF (MOD(NODE,2).EQ. 0) THEN
  CALL EIGT (1, NODE)
  CALL EIST (1, NODE)
  CALL EIMES (5, 2, 'NODE = X(I1). The number of mesh nodes, NODE, must be an odd integer greater than 4.')
ELSE IF (MOD(NODE,2).EQ. 0) THEN
  CALL EIGT (1, NODE)
  CALL EIST (1, NODE)
  CALL EIMES (5, 3, 'NODE = X(I1) must be an odd integer greater than 4.')
END IF
IF (ALPHA LE. 0.0) THEN
  CALL EIGT (1, ALPHA)
  CALL EIST (1, ALPHA)
  CALL EIMES (5, 4, 'ALPHA = Z(R1). The penalty weightings factor which controls smoothness, ALPHA, must be greater than 0.')
END IF
IF (MAXIT LE. 0.0) THEN
  CALL EIGT (1, MAXIT)
  CALL EIST (1, MAXIT)
  CALL EIMES (5, 5, 'MAXIT = X(I1). The maximum number of iterations, MAXIT, must be greater than 0.')
END IF
IF (BND(1).GT. BND(2)) THEN
  CALL EIGT (1, BND(1))
  CALL EIST (1, BND(1))
  CALL EIMES (5, 6, 'BND(1) = Z(R1) and BND(2) = Z(R2). The minimum value for X, BND(1), must be less than or equal to the maximum value for X, BND(2).')
END IF
IF (INIT.NE. 0) THEN
  CALL EIGT (1, NE.O) OR, DENS(NODE).NE.O) THEN
  CALL EIST (1, DENS(1))
  CALL EIGT (2, DENS(NODE))
  CALL EIST (2, DENS(NODE))
  CALL EIMES (5, 7, 'DENS(1) = X(R1) and DENS(NODE)=X(I1) must be less than or equal to the density must be zero.')
END IF
IF (ISMIN(NODE,DENS,1)).LT. 0) THEN
  CALL EIMES (5, 8, 'The initial estimates of the density, DENS, must be greater than or equal to 0.')

```



ORIGINAL PAGE IS  
OF POOR QUALITY

```

END IF
END IF
NOB1 = 0
DO 10 I=1, NOBS
  IF (X(I).LT.BNDS(1)) .OR. (X(I).GT.BNDS(2)) THEN
    NOB1 = NOB1 + 1
  END IF
10 CONTINUE
IF (NOB1.EQ. NOBS) THEN
  IF CALL EIMS (5, 9, 'All elements in X lie outside the interval BNDS(1) to BNDS(2). At least one element of X must lie in this interval.')
  END IF
  IF (EPS.LE. 0.0) THEN
    IF EPS1 = 1.0E-4
  ELSE EPS1 = EPS
  END IF
  IF (NIRCD(0).NE. 0) GO TO 2000
  Initialization
  Set initial densities
  C
  IMPTR = 0
  IF (INIT.EQ. 0) THEN
    DENS(1) = 0.0
    DENS(2) = 2.0/(BNDS(2)-BNDS(1))
    DENS(3) = 0.0
    N = 3
  ELSE
    M = NODE
  END IF
  Refine mesh
  20 IF (INIT.EQ. 0) THEN
    MOLD = M
    IMPTR = IMPTR + 1
    M = MING(NODE, MINCR(IMPTR))
  END IF
  Get mesh interval width
  H = (BNDS(2)-BNDS(1))/(M-1)
  H2 = H*H
  H3 = H*H*H
  Make initial DENS integrate to 1.
  IF (INIT.NE. 0) THEN
    CALL SSCAL (NODE, 1.0/(H*SSUM(NODE, DENS, 1)), DENS, 1)
  END IF
  Set mesh nodes
  B(1) = BNDS(1)
  DO 30 I=2, M
    B(I) = B(I-1) + H
  30 CONTINUE
  Set B indices for interpolating X
  IPTR = 0
  40 IPTR = IPTR + 1
  IF (X(IPTR).LT. BNDS(1)) GO TO 40
  DO 60 K=1, M-1
    ILOHI(K,1) = IPTR
    IF ILOHI(K,2) = IPTR
      IF (IPTR.LE. NOBS) THEN
        IF (X(IPTR).LT. B(K+1)) THEN
          ILOHI(K,2) = ILOHI(K,2) + 1
          IPTR = IPTR + 1
          IF (IPTR.LE. NOBS) GO TO 50
        END IF
      END IF
    END IF
  60 CONTINUE

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

10 FACTOR = 2.0*CK*PH*W3
C
C      Initialize mesh node densities
C      IF (INIT.EQ.0) THEN
C          Via DESPT
C          CALL DESPT (M-2, B(2), 1, MOLD, BNDS, DENS, DENEST, WK, WK)
C          &
C          TEMP = 1.0/(CK*MXH)
C          DO 30 I=2, M-1
C              DENS(I) = AMAX1(TEMP, SQRT(DENEST(I-1,1)))
C          DO 80 CONTINUE
C      ELSE
C          Via the initial estimates
C          DO 90 I=2, M-1
C              DENS(I) = SQRT(DENS(I))
C          DO 90 CONTINUE
C          DENS(M) = 0.0
C          Maximize
C          DO 140 ITER=1, MAXIT
C              Get Hessian - Lagrangian
C              HESS(1,1) = 0.0
C              HESS(1,2) = 0.0
C              HESS(2,1) = 0.0
C              BSMALL = 0.0
C              SUM = 0.0
C              CK** are true estimates = FK**2
C              DO 120 K=2, M-1
C                  KM1 = K-1
C                  KM2 = MAX0(1, K-2)
C                  KP1 = K+1
C                  KP2 = MIN0(M, K+2)
C                  FK = DENS(K)
C                  FK*1 = DENS(KM1)
C                  FK*2 = DENS(KM2)
C                  CK*1 = FK**2
C                  CK*2 = FK**2
C                  CK*1 = FK**2
C                  CK*2 = FK**2
C                  CK*1 = DENS(KP1)**2
C                  CK*2 = DENS(KP2)**2
C                  BK = B(K)
C                  BK*1 = B(KM1)
C                  SUM = SUM + CK
C                  TEMP(K,1) = HESS(1,1)*KM1 - 4.0*FK*CK*2*FACTOR
C                  SUM1 = 0.000
C                  SUM2 = 0.000
C                  SUM3 = 0.000
C                  DO 100 I=1, LOHI(K,1), ILOHI(K,2)
C                      TEMP = (X(I)-BK)/H
C                      CONS = (1.0-TEMP)/(CK+(CKP1-CK)*TEMP)
C                      SUM1 = SUM1 - CONS
C                      SUM2 = SUM2 + CONS*CONS
C                      SUM3 = SUM3 + CONS*(1.0-CONS)/TEMP
C                  DO 100 CONTINUE
C                  CK*CH1 = CK - CK*1
C                  DO 110 I=1, LOHI(KM1,1), ILOHI(KM1,2)
C                      CONS = (X(I)-BK*1)/H
C                      TEMP = CK*1 + CK*CH1*CONS
C                      SUM1 = SUM1 - CONS/TEMP
C                      SUM2 = TEMP*TEMP
C                      SUM3 = SUM3 + (CONS*CONS)/TEMP
C                  DO 110 CONTINUE
C                  TEMP = FACTOR*(CK*2+CK*2-4.0*(CK*1+CK*1)+6.0*CK) + SUM1
C                  TEMP = 2.0*TEMP
C                  BSMALL = BSMALL + 2.0*CK*TEMP

```

```

      HESS(3,M-1) = TEMP - 4.0*CKK*(5.0*FACTOR*SUM2)
      IF (K.NE. 2) HESS(3,KM1) = 4.0*FK*FKM1*(-4.0)*FACTOR*(SUM3)
      DENEST(KM1,1) = FK*TEMP
      DENEST(KM1,2) = -2.0*FK
      BSMALL = 1.0/H - SUM + BSMALL
      CALL SCOPY (M-2, DENEST(1,2), 1, DENEST(1,3), 1)
      CALL SADD (M-2, -BSMALL/(2.0*SUM), HESS(3,1), LDHESS)
      CALL SCOPY (M-4, HESS(1,3), LDHESS, HESS(5,1), LDHESS)
      HESS(5,M-3) = 0.0
      HESS(5,M-2) = 0.0
      CALL SCOPY (M-3, -HESS(2,2), -LDHESS, HESS(4,1), LDHESS)
      HESS(4,M-2) = 0.0
      CALL L2IRB (M-2, HESS, LDHESS, 2, 2, HESS, LDHESS, 1, 1)
      CALL L2SRB (M-2, HESS, LDHESS, 2, 2, HESS, LDHESS, 1, 1)
      CALL L2FSRB (M-2, HESS, LDHESS, 2, 2, IPVT, DENEST, 1, 1)
      CALL L2FSRB (M-2, HESS, LDHESS, 2, 2, IPVT, DENEST, 1, 1)
      IF (NIRCD(1).NE. 0) GO TO 7000
      CONS = SDOOT(M-2, DENEST(1,3), 1, DENEST(1,2), 1)
      CONS = (1.0/H-SUM-SDOOT(M-2, DENEST(1,3), 1, DENEST(1,1), 1))/CONS
      CALL SAXPY (M-2, CONS, DENEST(1,2), 1, DENEST(1,1), 1)
      CALL SAXPY (M-2, -1.0, DENEST(1,1), 1, DENEST(1,1), 1)
      CALL SAXPY (M-2, -1.0, DENEST(1,1), 1, DENEST(1,1), 1)
      TEMP = SNRM2(M-2, DENEST(2), 1)
      IF (SNRM2(M-2, DENEST, 1) .LT. EPSI*TEMP) GO TO 150
      TEMP = TEMP*1.0E-4/SQRT(M-2.0)
      DO 130 I=2, M-1
      DENEST(I) = AMAX1(TEMP, DENEST(I))
      CONTINUE
      CALL ELSSII (1, MAXII)
      CALL ELHES (3, 1, 1, The maximum number of iterations '//
      (MAXIT=2*(II)) was exceeded.')
      CALL SHPROD (M-2, DENEST(2), 1, DENEST(2), 1, DENEST(2), 1)
      IF (M.NE. NODE) GO TO 20
      SUM1 = 0.0
      Evaluate log likelihood and penalty
      Penalty
      DO 140 K=1, M
      KM1 = MAX0(K-1, 1)
      KP1 = MIN0(K+1, M)
      SUM1 = SUM1 + (DENEST(KM1)-2.0*DENS(K))*(DENS(KP1))**2
      CONTINUE
      SUM2 = 0.0
      DO 170 I=1, NOBS
      IF (X(I).GE.BNDS(1) .AND. X(I).LE.BNDS(2)) THEN
      CALL D2SPT (1, X(I), 1, NODE, BNDS, DENS, DENEST, WK, WK,
      SUM2 = SUM2 + ALOG(DENEST(1,1))
      END IF
      CONTINUE
      STAT(1) = SUM2
      Evaluate M.L.P.E. mean and variance

```

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

```

SUM1 = 0.0
SUM2 = 0.0
DO 130 K=1, M-1
  FN = DENS(K)
  FKPI = DENS(K+1)
  BK = B(K)
  CONS = FK + FKPI
  TEMP = CONS + H2*TEMP/6.0 + 0.5*H*BK*CONS
  SUM1 = SUM1 + H2*TEMP/12.0 + H2*BK*TEMP/3.0 +
  SUM2 = SUM2 + H3*TEMP*FKPI/12.0 + H2*BK*TEMP/3.0 +
  SUM2 = 0.5*H*BK*BK*CONS
130 CONTINUE
STAT(3) = SUM1
STAT(4) = SUM2 - SUM1*SUM1
C 9000 CALL EIPDP ('DISPL ')
RETURN
END
/EOF

```



300.0000	45.0000
3.0000	0.3000
500.0000	25.0000
5.0000	0.5000
250.0000	12.5000
150.0000	7.5000
150.0000	0.0150
0.5000	75.0000
1500.0000	0.5000
20.0000	25.0000
350.0000	12.5000

1.0E-05 30

```

=====
Digital Equipment Corporation - VAX/VMS Version 04
=====

```

[illegible][illegible]

28

A 10x10 grid of dots forming a complex pattern. The pattern consists of a central 4x4 square of dots, with additional dots forming a larger square around it. The dots are arranged in a way that creates a series of nested squares and lines.

File DBAQ: [IRANDM33 CPR/1 (383,943,0), last revised on 23-NOV-1988 11:26, is a 33 block sequential file owned by UIC C11.111. The records are variable length with FORTRAN (FTN) carriage control. The longest record is 120 bytes.

Job: RANDM33 (689) queued to SYS688PRI on 23-NOV-1988 11:27 by user NETNOMPRIV, UIC-C11.111, under account 20100ADD at priority 100. started on printer \_TIF7: on 23-NOV-1988 11:27 from queue TIF7.

[illegible]

[illegible]



30

X AXIS PDF, CDF PLOT

31

ORIGINAL PAGE IS  
OF POOR QUALITY

SCDF PARAMETERS			
OF LOG OF CURRENT CYCLES, LOG XNM,	Y AXIS OF PDF, CDF PLOT	Y AXIS OF PDF, CDF PLOT	Y AXIS OF PDF, CDF PLOT
0.5723E+01	0.5919E+01	0.311E+01	0.399E+01
0.7023E+01	0.589E+01	0.339E+01	0.399E+01
0.7812E+01	0.7873E+01	0.329E+01	0.344E+01
0.8692E+01	0.889E+01	0.324E+01	0.344E+01
0.9538E+01	0.9051E+01	0.324E+01	0.344E+01
0.0000E+00	0.247E+02	0.149E+01	0.311E+01
0.1291E+00	0.2020E+00	0.339E+01	0.399E+01
0.4172E+00	0.3251E+00	0.329E+01	0.344E+01
0.9232E+00	0.4430E+00	0.324E+01	0.344E+01
0.1000E+01	0.9430E+00	0.324E+01	0.344E+01
0.5723E+01	0.5919E+01	0.311E+01	0.399E+01
0.7023E+01	0.589E+01	0.339E+01	0.399E+01
0.7812E+01	0.7873E+01	0.329E+01	0.344E+01
0.8692E+01	0.889E+01	0.324E+01	0.344E+01
0.9538E+01	0.9051E+01	0.324E+01	0.344E+01
0.0000E+00	0.247E+02	0.149E+01	0.311E+01
0.1291E+00	0.2020E+00	0.339E+01	0.399E+01
0.4172E+00	0.3251E+00	0.329E+01	0.344E+01
0.9232E+00	0.4430E+00	0.324E+01	0.344E+01
0.1000E+01	0.9430E+00	0.324E+01	0.344E+01
0.5723E+01	0.5919E+01	0.311E+01	0.399E+01
0.7023E+01	0.589E+01	0.339E+01	0.399E+01
0.7812E+01	0.7873E+01	0.329E+01	0.344E+01
0.8692E+01	0.889E+01	0.324E+01	0.344E+01
0.9538E+01	0.9051E+01	0.324E+01	0.344E+01
0.0000E+00	0.247E+02	0.149E+01	0.311E+01
0.1291E+00	0.2020E+00	0.339E+01	0.399E+01
0.4172E+00	0.3251E+00	0.329E+01	0.344E+01
0.9232E+00	0.4430E+00	0.324E+01	0.344E+01
0.1000E+01	0.9430E+00	0.324E+01	0.344E+01
0.5723E+01	0.5919E+01	0.311E+01	0.399E+01
0.7023E+01	0.589E+01	0.339E+01	0.399E+01
0.7812E+01	0.7873E+01	0.329E+01	0.344E+01
0.8692E+01	0.889E+01	0.324E+01	0.344E+01
0.9538E+01	0.9051E+01	0.324E+01	0.344E+01

Digital Equipment - VAX/VMS version 4.

[illegible]

File DBAO:[PLOT1.CPR:1 (359,209,0)], last revised on 23-NOV-1988 11:26, is a 2 block sequential file owned by UIC 211,111. The records are variable length with FORTRAN (FTN) carriage control. The longest record is 25 bytes.

Job PLOT1 (687) queued to SY5688PRT on 23-NOV-1988 11:26 by user-NETNONPRIV, UIC 211,111, under account 2010CADD-se priority-100 started on printer TTF7 on 23-NOV-1988 11:26 from queue TTF7.

[illegible]

[illegible]



ORIGINAL PAGE IS  
OF POOR QUALITY

(E12. 4, 1X, E12. 4)  
0. 3723E+01 0. 0000E+00  
0. 3919E+01 0. 3477E-02  
0. 5113E+01 0. 1613E-01  
0. 5311E+01 0. 4149E-01  
0. 5306E+01 0. 8113E-01  
0. 5702E+01 0. 1391E+00  
0. 5898E+01 0. 2030E+00  
0. 7074E+01 0. 3773E+00  
0. 7289E+01 0. 3638E+00  
0. 7483E+01 0. 4523E+00  
0. 7481E+01 0. 5419E+00  
0. 7876E+01 0. 6291E+00  
0. 8072E+01 0. 7110E+00  
0. 8268E+01 0. 7852E+00  
0. 8444E+01 0. 8494E+00  
0. 8637E+01 0. 9023E+00  
0. 8835E+01 0. 9430E+00  
0. 9031E+01 0. 9716E+00  
0. 9246E+01 0. 9893E+00  
0. 9442E+01 0. 9978E+00  
0. 9638E+01 0. 1000E+01

## 8.0 APPENDIX C

### RANDOM4 SAMPLE PROBLEM: SOURCE, INPUT AND OUTPUTFILES



00000000000000000000

[illegible]

38

```

JOB,JMS00MFF,US49A0530,RT=50,F=30,MFL=200000000
ACCELTE,UFW=L0LAR.
DELETE,PON=NR4BLD,ID=SMBOYCE.
C C
REWIND,DN=$BLD.
3UILL,EQ=DEL,PON=NR4BLD,ID=SMBOYCE.
DELETE,PON=NR4BLD,ID=SMBOYCE,ED=-1.
/EOF
C C CHAMIS MICROMECHANICS CONSTITUTIVE EQUATIONS:
INTEGRATED-AND-APPLIED TO FATIGUE STRENGTH.
INTEGER NTOT,ISEED,NIT,NLNM3,MAXI,NOD
REAL XM,XS,YM,YS,EPS,KWKSPI(999),ALPHA,
DIMENSION XLNMQ(10000),XLNMF(10000),SU(10000)
DIMENSION SIGO(10000),SIG(10000)
DIMENSION XNM(10000),XMX(10000), $\epsilon$ XQ(10000)
DIMENSION TET(10000),IBL(10000),I(10000)
DIMENSION STAT(9999)
DIMENSION RMDS(999),DENR(999)
DIMENSION CM(999),FP(999)
DIMENSION SM(10)
DIMENSION XP(1),CUM(1)
DIMENSION AL(12)
COMMON/MEP1/KPRINT,TOL,MAXFN
KPRINT=1
TOL=1.0E-06
MAXFN=50
1001 FORMAT(5E12.4)
1002 FORMAT(E12.4,2X,I4)
1003 FORMAT(I4,I4)
1004 FORMAT(I4)
1005 FORMAT(I12,I12)
1006 FORMAT(E12.4)
C LOGNORMAL ULTIMATE TENSILE STRENGTH, SF
READ(5,1005) ISEED,NTOT
WRITE(6,1005) ISEED,NTOT
XM=X*1000.
XS=X/45.
READ(5,1006) XM,XS
WRITE(6,1006) XM,XS
YS = SORT(LOG(1+.0+(XS/XM)**2))
YS = SORT(LOG(1+.0+(XS/XM)**2))
YM = LOG(XM) - 0.5*YS**2.
CALL RNSET(ISEED)
CALL RNLNLT(NTOT,YM,YS,SF)
WRITE(6,1001)USE(I,I=1,NTOT)
WRITE(6,2020)
2020 FORMAT('LOGNORMAL SF')
WRITE(6,1001)(SF(I),I=1,NTOT)
C LOGNORMAL LOG OF FINAL CYCLE, XLNMF
WRITE(6,1005) ISEED,NTOT
READ(5,1006) XM,XS
WRITE(6,1006) XM,XS
XM=X/8.
XS = 0.8
YS = SORT( LOG(1.0+(XS/XM)**2) )
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNLNLT(NTOT,YM,YS,XLNMF)
WRITE(6,19,1001) (XLNMF(I),I=1,NTOT)
WRITE(6,2021)
2021 FORMAT('LOGNORMAL XLNMF')
WRITE(6,1001) (XLNMF(I),I=1,NTOT)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

C LOGNORMAL FATIGUE STRENGTH AT REFERENCE CONDITIONS, S0
WRITE(6,1005) ISEED,NTOT
READ(5,1006) XM,XS
WRITE(6,1006) XM,XS
XM = 500.
XS = 25.
YS = SQRT( LOG(1.0+(XS/XM)**2) )
YM = LOG(XM) - 0.5*YS**2
CALL RNSET( ISEED )
CALL RNLNL( NTOT, YM, YS, S0 )
WRITE(6,2022)
WRITE(20,1001) (S0(I), I=1, NTOT)
2022 FORMAT( ' LOGNORMAL S0' )
C LOGNORMAL LOG-OF-REFERENCE CYCLES- XLNMO.
WRITE(6,1005) ISEED,NTOT
READ(5,1006) XM,XS
WRITE(6,1006) XM,XS
XM = 7.
XS = 0.7
YS = SQRT( LOG(1.0+(XS/YM)**2) )
YM = LOG(XM) - 0.5*YS**2
CALL RNSET( ISEED )
CALL RNLNL( NTOT, YM, YS, XLNMO )
WRITE(6,2023)
WRITE(21,1001) (XLNMO(I), I=1, NTOT)
2023 FORMAT( ' LOGNORMAL XLNMO' )
C LOGNORMAL FATIGUE STRENGTH AT CURRENT CONDITIONS, S
WRITE(6,1005) ISEED,NTOT
READ(5,1006) XM,XS
WRITE(6,1006) XM,XS
XM = 250.
XS = 12.5
YS = SQRT( LOG(1.0+(XS/XM)**2) )
YM = LOG(XM) - 0.5*YS**2.
CALL RNSET( ISEED )
CALL RNLNL( NTOT, YM, YS, S )
WRITE(6,2024)
WRITE(22,1001) (S(I), I=1, NTOT)
2024 FORMAT( ' LOGNORMAL S' )
C DEFINE RANDOM STRESSES
C LOGNORMAL REFERENCE STRESS, SIG0
WRITE(6,1005) ISEED,NTOT
READ(5,1006) XM,XS
WRITE(6,1006) XM,XS
XM = 20.
XS = 1.
YS = SQRT( LOG(1.0+(XS/XM)**2) )
YM = LOG(XM) - 0.5*YS**2.
CALL RNSET( ISEED )
CALL RNLNL( NTOT, YM, YS, SIG0 )
C CHANGE SIG0 TO NEGATIVE VALUES FOR COMPRESSIVE
C RESIDUAL STRESSES
DO 401 I = 1, NTOT
  SIG0(I) = -SIG0(I)
401 CONTINUE
WRITE(26,1001) (SIG0(I), I=1, NTOT)
2036 FORMAT( ' LOGNORMAL SIG0' )
WRITE(6,1001) (SIG0(I), I=1, NTOT)
C LOGNORMAL CURRENT STRESS, SIG
WRITE(6,1005) ISEED,NTOT

```

```

C C
READ(5,106)XM,XS
WRITE(5,1006)XM,XS
XS=7.5
YS=SQRT(LOG(1.0+(XS/XM)**2.))
CALL LOG(XM),0.5*YS**2.
CALL RNSET(ISEED)
CALL RNLN(NTOT),YM,YS,SIG)
WRITE(6,2037)
WRITE(6,2037)
WRITE(6,2037)
WRITE(6,1001)(SIG(I),I=1,NTOT)
WRITE(6,1001)(SIG(I),I=1,NTOT)
NORMAL EXPONENTS, XXN,XXM,XXQ
YM=0.5
YS=0.515
WRITE(6,1005)ISEED,NTOT
READ(5,1006)YM,YS
WRITE(6,1006)YM,YS
CALL RNSET(ISEED)
CALL RNIN(NTOT),XXN
DO 101 I=1,NTOT
XXN(I)=YS*XXN(I)+YM
CONTINUE
101 WRITE(6,2035)
WRITE(6,2035)
FORMAT(6,'NORMAL XXN')
WRITE(6,1001)(XXN(I),I=1,NTOT)
WRITE(6,1001)(XXN(I),I=1,NTOT)
CALL RNSET(ISEED)
CALL RNIN(NTOT),XXM
DO 201 I=1,NTOT
XXM(I)=YS*XXM(I)+YM
CONTINUE
201 WRITE(6,2036)
WRITE(6,2036)
FORMAT(6,'NORMAL XXM')
WRITE(6,1001)(XXM(I),I=1,NTOT)
WRITE(6,1001)(XXM(I),I=1,NTOT)
CALL RNSET(ISEED)
CALL RNIN(NTOT),XXQ
DO 301 I=1,NTOT
XXQ(I)=YS*XXQ(I)+YM
CONTINUE
301 WRITE(6,2037)
WRITE(6,2037)
FORMAT(6,'NORMAL XXQ')
WRITE(6,1001)(XXQ(I),I=1,NTOT)
WRITE(6,1001)(XXQ(I),I=1,NTOT)
C DEFINE DETERMINISTIC TEMPERATURES
TF=1500.
C C C C
ID=30.
I=30.
C NORMAL TEMPERATURES, IF, IO, I
C NORMAL FINAL (HEATING) TEMPERATURE, I
WRITE(6,1005)ISEED,NTOT
READ(5,1006)YM,YS
WRITE(6,1006)YM,YS
YM=1500.
YS=75.
CALL RNSET(ISEED)
CALL RNIN(NTOT),IF)
DO 405 I=1,NTOT
TF(I)=YS*IF(I)+YM
CONTINUE
405 WRITE(6,2046)

```

[illegible]

ORIGINAL PAGE IS  
OF POOR QUALITY

```

2020 FORMAT (10, SORTED LOG OF CYCLES)
WRITE (6,1001) (XNM(I), I=1, NTOT)
C CALCULATE PDF OF LOG OF CURRENT CYCLES, LOG XNM
C USING THE MAXIMUM ENTROPY METHOD
C CALCULATE SAMPLE MOMENTS, SM
C NUMBER OF MOMENTS, MMH=4
CALL SMOM(XNM, MMH, NTOT, SM)
WRITE (6,1001) (SM(I), I=1, MMH)
WRITE (6,2038)
2038 FORMAT (1, SAMPLE MOMENTS)
C OBTAIN MAXIMUM ENTROPY DISTRIBUTION
KSTART=1
KDATA=1
C CALCULATE MAX AND MIN ORDINATES FOR PDF (AND CDF)
BND1(1) = XNM(1) - 0.05*XNM(1)
BND2(2) = XNM(NTOT) + 0.05*XNM(NTOT)
WRITE (6,8877) BND1(1), BND2(2)
WRITE (6,8877) BND1(1), BND2(2)
2039 FORMAT (1, BND1(1), BND2(2), E12.4, 1X, E12.4)
CALL ME1(XNM, SM, BND1(1), BND2(2), 0.0, KSTART, KDATA, PL, CUM)
WRITE (6,1001) (AL(I), I=1, MMH+1)
WRITE (6,2039)
2039 FORMAT (1, LAGRANGIAN MULTIPLIERS)
C CALCULATE VALUES OF ORDINATES FOR PDF (AND CDF)
C NUMBER OF ORDINATES USED
C CALCULATE WINDOW WIDTH, HH
NODE=21
HH=(BND2(2)-BND1(1))/(NODE-1)
C CALCULATE VALUES OF LOG OF CURRENT CYCLES AT WHICH PDF IS ESTIMATED;
C ALSO CALLED "NODE" VALUES
DO 6001, I=1, NODE-2
  BND1(I+2)=BND1(1) + (I*HH)
  BND2(I+2)=BND2(1) + (I*HH)
6001 CONTINUE
983 FORMAT (1, LOG OF CURRENT CYCLES, LOG XNM)
WRITE (6,1001) (BND1(I), I=1, NODE)
C REORDER BND1 FOR PLOTTING
C
SAVE1 = BND1(2)
SAVE2 = BND1(NODE)
BND1(NODE)=BND1(2)
BND1(2)=BND1(NODE)
DO 8002, I=1, NODE-2
  BND1(I+1)=BND1(I+2)
  BND2(I+1)=BND2(I+2)
8002 CONTINUE
BND1(NODE-1)=SAVE2
BND2(NODE-1)=SAVE1
WRITE (6,984)
984 FORMAT (1, ORDERED LOG OF CURRENT CYCLES, LOG XNM,
1X, AXIS, PDF, CDF, PLOT)
WRITE (6,1001) (BND1(I), I=1, NODE)
C CALCULATE VALUES OF THE PDF AT EACH ORDINATE
DO 1000 I=1, NODE
  C FOR 4 MOMENTS THERE ARE 5 LAGRANGIAN MULTIPLIERS
  DENS(I)=EXP(AL(I)+AL(2)*BND1(I)+AL(3)*BND1(I)**2
1+AL(4)*BND1(I)**3+AL(5)*BND1(I)**4)
1000 CONTINUE
C WRITE LOG OF CURRENT CYCLES AND PDF OF LOG OF CURRENT CYCLES,

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

C LOG-XNM TO PLOT FILES
WRITE(34,990)
990 FORMAT('E12.4,1X,E12.4')
991 FORMAT('E12.4,1X,E12.4')
C CALCULATE CDF OF LOG OF CURRENT CYCLES
IOPT=2
C
READ(3,1004)IOPT
WRITE(6,992)
992 FORMAT(' SCDF PARAMETERS')
WRITE(6,1004)IOPT
XO=BNDX(1)
DO 5003,I=1,NODE
P=GCDF(XO,IOPT,NODE,BNDX,DENS)
BNDX(I)=XO
XO=XO*HH
XO=XO*HH
DISX(I)=P
6003 CONTINUE
WRITE(6,994)
994 FORMAT(' CDF OF LOG OF CURRENT CYCLES, LOG XNM,
1Y AXIS OF PDF, CDF PLOT')
WRITE(6,1001)(DISTX(I),I=1,NODE)
C
WRITE(6,993)
993 FORMAT(' ORDERED LOG-OF CURRENT CYCLES, LOG XNM,
1X AXIS OF PDF, CDF PLOT')
WRITE(6,1001)(BNDX(I),I=1,NODE)
WRITE(6,1001)(BNDX(I),I=1,NODE)
C
WRITE LOG OF CURRENT CYCLES AND CDF OF LOG OF CURRENT
CYCLES TO THE PLOT FILES
WRITE(35,990)
WRITE(35,991)(BNDX(J),DISTX(J),J=1,NODE)
STOP
END
C
SUBROUTINE SORT (Y,N)
DIMENSION Y(10000)
C Y IS THE ARRAY TO BE SORTED
C AT COMPLETION Y(1) IS SMALLEST VALUE
C AT COMPLETION Y(N) IS LARGEST VALUE
N1 = N - 1
DO 1 I=1,N1
J = I + 1
DO 2 K=J,N
IF (Y(I).LT.Y(K))GO TO 2
TEMP = Y(I)
Y(I) = Y(K)
Y(K) = TEMP
2 CONTINUE
1 CONTINUE
RETURN
END
C
SUBROUTINE SHOM(X,M,NSAMP,SH)
C CALCULATES SAMPLE CENTRAL MOMENTS
C X(I) = SAMPLE VALUES, DIMENSION NSAMP
C M = NUMBER OF MOMENTS DESIRED
C NSAMP = SAMPLE SIZE
C SH = VALUE OF MOMENTS, DIMENSION M
DIMENSION X(10000),SH(10)

```

```

1- CALCULATE MEAN
SUM=0.0
DO 1 I=1,NSAMP
  SUM=SUM+X(I)
1 SUM=SUM/NSAMP
IF (M.LT.3) RETURN
2- CALCULATE VARIANCE
SUM=0.0
DO 2 I=1,NSAMP
  SUM=SUM+(X(I)-SUM(1))**2
2 SUM=SUM/NSAMP
IF (M.LT.3) RETURN
3- CALCULATE HIGHER MOMENTS
DO 3 I=3,M
  SUM=0.0
DO 3 J=1,NSAMP
  SUM=SUM+(X(J)-SUM(1))**I
3 SUM=SUM/NSAMP
4- CONTINUE
  RETURN
END

SUBROUTINE MEPI(N,CM,XMIN,XMAX,NXP,XP,KSTART,KDATA,AL,CUM)
  IMPLICIT REAL*8 (A-H,O-Z)
  EXECUTIVE PROGRAM FOR USING MAXIMUM ENTROPY METHOD CONSTRAINED BY
  MOMENTS TO GENERATE A DENSITY FUNCTION

  DIMENSION AL(*), CM(*), ETA(4), XP(*), CUM(*), CC(3),ALE(10)
  COMMON /FAIL, NFAIL, XX(16,101), C(8),M
  COMMON /HELP, S(101), X(16,101), C(8),M
  ABOVE LINE DIFFERENT FROM TEXT
  COMMON /MEPI, KPRINT, TOL, MAXFN
  DATA KPRINT, TOL, MAXFN / 1, 1.E-6, 70 /
  IF (N.EQ.1) KSTART=2

  WRITE THE INPUT DATA
  IF (KDATA.EQ.0) GO TO 1
  WRITE (4,24)
  WRITE (6,25) KDATA
  WRITE (6,26) KPRINT
  WRITE (6,28) N
  WRITE (6,29) XMAX
  WRITE (6,30) XMIN
  WRITE (6,31) (CM(I), I=1,4)
  IF (N.GT.4) WRITE (6,21) (CM(I), I=5,N)
  IE (ABS(CM(1)), 11, 1.E-4) GO TO 48
  WRITE (6,32) TOL
  WRITE (6,33) NXP
  CONTINUE
  NFAIL=0
  N=31
  X2MIN=0.0
  X2MAX=1.
  SAVE CH
  DO 100 I=1,N
    CC(I)=CM(I)
  100 CC
  CALCULATE THE MOMENTS AT THE MODIFIED LIMITS
  CALL TRN1 (XMAX,XMIN,CC,X2MAX,X2MIN,N)
  CALCULATE THE MOMENTS ABOUT THE ORIGIN FOR THE MODIFIED LIMITS
  STORE THEM IN COMMON IN C

```



```

C      CALL CONVER(CC,N)
C
C      GENERATE THE SIMPSON MULTIPLIERS AND STORE THEM IN HELP COMMON
C
C      CALL SIMSON
C
C      GENERATE THE X,S POWER FOR SUBROUTINE FUNCT, STORE THEM IN HELP
C      COMMON ARRAY
C
C      CALL MULTI-(X2MAX,X2MIN,N)
C
C      DEFINE THE INPUT DATA FOR SUBROUTINE MPOPT
C
C      ETA(1)=1.0-12
C      ETA(2)=TOL
C      ETA(3)=1.0-34
C      ETA(4)=1.0-34
C      MODE=1
C      UMIN=0.0
C
C      WRITE THE INTERMEDIATE RESULTS YOU HAVE OBTAINED SO FAR
C
C      IF (KPRINT.EQ.0) GO TO 2
C      WRITE (6,34)
C      WRITE (6,35) M
C      WRITE (6,36) X2MAX,X2MIN
C      WRITE (6,37) (CC(I),I=1,4)
C      IF (N.GT.5) WRITE (6,22) (CC(I),I=5,N)
C      WRITE (6,38) (C(I),I=1,4)
C      IF (N.GT.4) WRITE (6,22) (C(I),I=5,N)
C      WRITE (6,39) (ETA(I),I=1,4)
C      CONTINUE
C
C      FIND A STARTING POINT FOR SUBROUTINE MPOPT TO START THE OPTIMIZATION ALGORITHM
C
C      IF (KSTART.EQ.0) GO TO 16
C      IF (KSTART.EQ.4) WRITE (6,44)
C      CALL START (X2MAX,X2MIN,AL,KSTART,CC,N,KPRINT,UMIN,MODE,MAXFN,ETA)
C      IF (NFAIL.EQ.1) GO TO 9
C
C      PRINT THE STARTING VALUES
C
C      IF (KPRINT.EQ.0) GO TO 7
C      GO TO (3,4,5,6), KSTART
C      WRITE (6,40)
C      WRITE (6,41) (AL(I),I=1,4)
C      IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
C      GO TO 7
C      WRITE (6,42)
C      WRITE (6,41) (AL(I),I=1,4)
C      IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
C      GO TO 7
C      WRITE (6,43)
C      WRITE (6,41) (AL(I),I=1,4)
C      IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
C      GO TO 7
C      WRITE (6,44)
C      WRITE (6,41) (AL(I),I=1,4)
C      IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
C      GO TO 7
C      CONTINUE
C
C      RANGE=MAX-XMIN
C      ..... CHANGE STARTING VALUES TO 0-1 DOMAIN FOR KSTART=0

```

ORIGINAL PAGE IS  
OF POOR QUALITY

46

ORIGINAL PAGE IS  
OF POOR QUALITY

```

C..... THIS ALGORITHM IS SENSITIVE TO THE WAY THE OPERATORS TO THE RIGHT
C NUMERICAL RESULTS
NPL=N+1
IF (ABS(XMIN).LT.1.E-10) GO TO 19
DO 17 I=2,NPL
  ALS(I)=0.0
  I1=I-1
  DO 18 J=I1,N
    ALS(I)=ALS(I)+FACTO(J)*XMIN**(J-I1)*RANGE**I1*AL(J+1)/FACTO(I1)
  1/FACTO(J-I1)
  18 CONTINUE
  17 CONTINUE
  GO TO 50
DO 20 I=2,NPL
  ALS(I)=RANGE**(I-1)*AL(I)
C..... PUT AL(I) IN PROPER LOCATIONS
DO 51 I=1,N
  AL(I)=ALS(I+1)
51 CONTINUE
7 CONTINUE
NFAIL=0
IF (KPRINT.EQ.0) GO TO 3
WRITE (6,45)
CONTINUE
8 AL(N+1)=2.
  AL(N+2)=0.0
  CALL MPORT (AL,N,ETA,UMIN,MAXEN,MODE,KPRINT)
  IF (NFAIL.EQ.0) GO TO 10
  IF (KSTART.EQ.4) GO TO 9
C THE PROGRAM HAS FAILED SO FAR, TRY ANOTHER STARTING POINT AND TRY
C AGAIN
KSTART=KSTART+1
IF (KSTART.EQ.4.AND.N.LE.2) GO TO 9
GO TO 2
9 CONTINUE
WRITE (6,46)
CALL EXIT
10 CONTINUE
C CALCULATE THE ZEROth LAGRANGIAN MULTIPLIER
SUM=0.0
DO 12 I=1,M
  SZ=0.0
  DO 11 K=1,N
    SZ=SZ+AL(K)*XX(K,I)
  11 CONTINUE
  SUM=SUM+SZ(I)*EXP(SZ)
  12 CONTINUE
  NPL=N+1
  DO 13 I=1,N
    K=N+2-I
    AL(K)=AL(K-1)
  13 CONTINUE
  DELTA=(X2MAX-X2MIN)/FLOAT(M-1)
  AL(1)=-ALOG(SUM*DELTA/3.)
  WRITE (6,101)UMIN
  101 FORMAT(26H SUM OF RESIDUALS SQUARED=,E12.5)
  IF (KPRINT.EQ.0) GO TO 14
  WRITE (6,47) (AL(I),I=1,NPL)
  14 CONTINUE
C..... RESET KSTART TO ZERO

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

0000      KSTART=0
0000      CALCULATE THE LAGRANGIAN MULTIPLIERS FOR THE ORIGINAL LIMITS
0000      CALL TRN2 (XMAX,XMIN,AL,X2MAX,X2MIN,N)
0000      CALCULATE THE CUMULATIVE DISTRIBUTION FUNCTION VALUE AT THE GIVEN
0000      POINT
0000      IF(NXP.EQ.0)RETURN
0000      DO 15 I=1,NXP
0000      CUM(I)=CDF(XMIN+XMAX*XP(I)-AL,NPL)
0000      CONTINUE
0000      RETURN
15
0000      FORMAT (57X,4E18.9,/)
0000      FORMAT (57X,4E18.9,/)
0000      FORMAT (1H1,/,20X,/, INPUT DATA FOR SUBROUTINE  MEP1,/,20X,33(' '
25      1),/,/)
0000      IF INPUT DATA IS PRINTED OUT FOR KDATA=1 ONLY  . . . KDATA =
26      1,/,/)
0000      IF INTERMEDIATE OUTPUT EVERY KPRINT(TH) CYCLE  . . . KPRINT =
28      1,/,/)
0000      IF NUMBER OF KNOWN FIRST MOMENTS  . . . . . N=
29      1,/,/)
0000      IF HIGHER LIMIT  . . . . . XMAX =
30      1,/,/)
0000      IF LOWER LIMIT  . . . . . XMIN =
31      1,/,/)
0000      IF FIRST MOMENTS  . . . . . CC(1) =
32      1,/,/)
0000      IF THE ALLOWED TOLERANCE IN LAGRANGIAN EQUATIONS  . . . TOL =
33      1,/,/)
0000      IF THE CUMULATIVE DISTRIBUTION REQUIRED AT NXP POINTS,NXP =
34      1,/,/)
0000      IF INPUT DATA,INTERMEDIATE RESULTS FOR SUBROUTINE  MEP1,2
35      10X,41(' '),/,/)
0000      IF NUMBER OF INTEGRATION STATION  . . . . . M =
36      1,/,/)
0000      IF MODIFIED MAXIMUM AND MINIMUM LIMITS  . . . X2MAX, X2MIN =
37      1,/,/)
0000      IF MODIFIED MOMENTS ABOUT THE EXPECTED VALUE  . . . CC(I) =
38      1,/,/)
0000      IF MODIFIED MOMENTS ABOUT THE ORIGIN  . . . . . C(I) =
39      1,/,/)
0000      IF SUBROUTINE MPOPT TOLERANCES  . . . . . ETA(I) =
40      1,/,/)
0000      IF NORMAL ASSUMPTION STARTING METHOD,34(' '),/,)  AL(I) =
41      1,/,/)
0000      IF STARTING VALUES  . . . . .
42      1,/,/)
0000      IF UNIFORM ASSUMPTION STARTING METHOD,35(' '),/,)
43      1,/,/)
0000      IF N-POINTS STARTING METHOD,35,/,/,)
44      1,/,/)
0000      IF STEP BY STEP STARTING METHOD,29(' '),/,)
45      1,/,/)
0000      IF CYC NUMF NORMGRAD  TOTAL,24X,VARIABLES,40
46      1X,RESIDUALS,/, NO,10X,RESIDUALS  R(1)  R(2)  R
47      3(4),/,/)
0000      IF THE PROGRAM HAS FAILED')
48      1,/,/)
0000      IF THE MODIFIED LAGRANGIAN MULTIPLIERS ARE  . . . . .
49      1,4E18.9,4E18.9)
0000      WRITE(6,19)
0000      FORMAT(53H WARNING - MEAN IS NEARLY ZERO AND MEP1 WILL NOT WORK/12
0000      1H TRANSFORM X)
0000      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

C
C
SUBROUTINE MPORT (X,NDIM,ETA,EST,MAX,MODE,IPRINT)
IMPLICIT REAL*8 (A-H,O-Z)
REAL* KTB,IPRINT
COMMON /FAIL/ NFAIL
DIMENSION Y(10), X1(10), X2(10), G1(10), G2(10), ALEA(10), H(10), P
1(10,10), Y(10), PY(10), PE(10), EIA(10), SIGU(10), RR(8)
EXTERNAL FUNCT
KRSI=0
KTB=0
N1=0
IFLAG=0
M=0
N2=NDIM+1
N1=NDIM+2
NUMF=0
IER=0
DO 1 I=1,N1
X1(I)=X(I)
CONTINUE
CALL FUNCT (NDIM,X1,F1,G1,RR)
NUMF=NUMF+1
DO 2 I=1,NDIM
X2(I)=X1(I)
G2(I)=G1(I)
H(I)=-G1(I)
CONTINUE
E2=1
X2(N2)=X1(N2)
X2(N1)=X1(N1)
CONTINUE
KOUNI=0
EPS=ETA(4)
CALL LINES (FUNCT,X2,H,RQ,NDIM,F2,G2,NUMF,IER,EPS,EST,RR)
IF (NFAIL.EQ.1) RETURN
IF (IER.NE.0) GO TO 10
DO 4 I=1,N1
RGV(I)=X2(I)
ALFA(I)=X2(I)
CONTINUE
RO=-RO
GG=0.
DO 5 I=1,NDIM
GG=GG+G2(I)*G2(I)
CONTINUE
GG=SQRT(GG)
IF (IPRINT.EQ.0) GO TO 7
IF (MOD(KTB,IPRINT).NE.0) GO TO 6
CALL OUTP (X2,F2,M,NDIM,GG,NUMF,RR)
KTB=KTB+1
DO 9 I=1,N1
DO 8 J=1,N1
P(I,J)=0.
CONTINUE
P(I,I)=1.
CONTINUE
PRINT*,KOUNT
KOUNT=0
DO 12 I=1,NDIM
Y(I)=G2(I)
PRINT*, GOT BY A1'
CONTINUE
Y(N2)=F2

```

```

X=N1-ETA1
U=0.
DO 13 I=1,NDIM
  V=V+X2(I)*G2(I)
  PRINT*, 'GOT BY A2'
  CONTINUE
YA=0.
DO 14 I=1,N1
  YA=YA+Y(I)*ALFA(I)
  PRINT*, 'GOT BY A3'
  CONTINUE
VYA=V-YA
RIGV(KOUNT)=V
DO 15 I=1,N1
  EY(I)=0.
  PE(I)=P(I,KOUNT)
  PY(I)=PY(I)+P(J,I)*Y(J)
  PRINT*, 'GOT BY A4'
  EY=PY(KOUNT)
  IF (ABS(EY).LT.ETA(3)) GO TO 31
  PY(KOUNT)=PY(KOUNT)-1.
DO 16 I=1,N1
  DO 16 J=1,N1
    P(I,J)=P(I,J)-PE(I)*PY(J)/EY
    PRINT*, 'GOT BY A5'
  DO 17 I=1,N1
    ALFA(I)=0.
    DO 17 J=1,N1
      ALFA(I)=ALFA(I)+P(I,J)*RIGV(J)
      PRINT*, 'GOT BY A6'
    DEL=0.
    DO 18 I=1,NDIM
      DEL=DEL+G2(I)*X2(I)-ALFA(I)
      PRINT*, 'GOT BY A7'
    CONTINUE
    IF (ABS(DEL).GT.ETA(4)) GO TO 19
    IF (JFLAG.EQ.1) RETURN
    IFLAG=1
    GO TO 31
  IF LAG=0
    DO 20 I=1,N1
      H(I)=X2(I)-ALFA(I)
      IF (DEL.GT.0) H(I)=-H(I)
      PRINT*, 'GOT BY A8'
    CONTINUE
    DO 21 I=1,NDIM
      X1(I)=X2(I)
      G1(I)=G2(I)
      PRINT*, 'GOT BY A9'
    CONTINUE
    F1=F2
    X1(N2)=X2(N2)
    X1(N1)=X2(N1)
    X2(N2)=ALFA(N2)
    X2(N1)=ALFA(N1)
    PRINT*, 'GOT BY A10'
    CALL LINES (FUNC1,X2,H,RO,NDIM,F2,G2,NUMF,IER,EPS,EST,RR)
    PRINT*, 'GOT BY A11'
    IF (NEAL.EQ.1) RETURN
    PRINT*, 'GOT BY A12'
    IF (IER.NE.0) GO TO 30
    PRINT*, 'GOT BY A13'
    F (DEL.GT.0) RO=-RO

```

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

```
PRINT*,GOT BY AI*
GG=0.
DO 22 I=1,NDIM
GG=GG+G2(I)*G2(I)
PRINT*,GOT BY AIS'
CONTINUE
GG=GG+G2(I)
KOUNT=KOUNT+1
M=M+1
IF (IPRINT.EQ.0) GO TO 23
IF (MOD(KIB,IPRINT).NE.0) GO TO 23
CALL OUTP(X2,F2,M,NDIM,GG,NUMF,RR)
PRINT*,GOT BY H'
CONTINUE
KIB=KIB+1
IF (MODE.EQ.2) GO TO 25
IF (PRINT*,GOT BY HA'
IF (M.GI.MAX) GO TO 30
PRINT*,GOT BY HB'
NSOL=0
DO 24 I=1,NDIM
IF (ABS(RR(I)).GT.ETA(2)) NSOL=1
PRINT*,GOT BY HC'
CONTINUE
PRINT*,GOT BY HD'
IF (NSOL.EQ.0) GO TO 26
PRINT*,GOT BY HE'
GO TO 29
PRINT*,GOT BY HF'
IF (LOGG.LI.ETA(1)).OR.(M.GI.MAX)) GO TO 26
PRINT*,GOT BY HG'
GO TO 29
PRINT*,GOT BY HH'
CONTINUE
PRINT*,GOT BY HI'
IF (IPRINT.EQ.0) GO TO 27
PRINT*,GOT BY HJ'
WRITE(6,33)
PRINT*,GOT BY I'
CALL OUTP(X2,F2,M,NDIM,GG,NUMF,RR)
PRINT*,GOT BY J'
DO 28 I=1,NDIM
X(I)=X2(I)
CONTINUE
EST=F2
NFAIL=0
RETURN
CONTINUE
PRINT*,KOUNT
PRINT*,GOT BY JA'
IF (KOUNT.LE.N1) GO TO 11
PRINT*,GOT BY JB'
GO TO 10
PRINT*,GOT BY JC'
PRINT 34, IER
NFAIL=1
RETURN
KRST=KRST+1
IF (KRST.GT.10) NFAIL=1
IF (NFAIL.EQ.1) RETURN
DO 32 I=1,NDIM
Y(I)=Y2(I)
G1(I)=G2(I)
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

32      CONTINUE
      F1=FC
      X1(N2)=X(N2)
      X1(N1)=X(N1)
      X2(N2)=X(N2)
      X2(N1)=X(N1)
      GO TO 3
33
34      FORMAT (//,1X, ' THE PROGRAM HAS FAILED---IER = ',I2)
      END
35
36      SUBROUTINE OUTP (XNEW,FQ,KOUNT,N1,GG,NUMF,R)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XNEW(*), R(*)
      WRITE (6,6) KOUNT,NUMF,BG,FQ,(XNEW(I),I=1,4),(R(I),I=1,4)
      IF (N1.LT.4) RETURN
      NN=N1-3
      GO TO (1,2,3,4,5), NN
      RETURN
      WRITE (6,7) XNEW(5),R(5)
      RETURN
      WRITE (6,8) (XNEW(I),I=5,6),(R(I),I=5,6)
      RETURN
      WRITE (6,9) (XNEW(I),I=5,7),(R(I),I=5,7)
      RETURN
      WRITE (6,10) (XNEW(I),I=5,8),(R(I),I=5,8)
      RETURN
37
38      FORMAT (1X,I3,I4,6E14.5,4E10.3)
      FORMAT (36X,5I4.5,4E10.3)
      FORMAT (36X,3E14.5,2E10.3)
      FORMAT (36X,3E14.5,14X,3E10.3)
      FORMAT (36X,4E14.5,4E10.3)
      END
39
40      SUBROUTINE LINES (FUNCT,X,H,AMBD,A,N,F,G,NUMF,IER,EPS,EST,RR)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 Z,DY, DY
      COMMON /FAIL/ NFAIL
      DIMENSION H(*), X(*), G(*), RR(*)
      IER=0
      DY=0.
      HNRH=0.
      GNRH=0.
      DO 1 J=1,N
      HNRH=HNRH+ABS(H(J))
      GNRH=GNRH+ABS(G(J))
      DY=DY+H(J)*G(J)
      PRINT*, 'GOT BY B1'
      CONTINUE
      IF (DY) 2,31,31
      PRINT*, 'GOT BY B2'
      IF (GNRH) 3,31,31
      PRINT*, 'GOT BY B3'
      ALFA=2.*(EST-F)/DY
      IF (X(N+1).GT.0.) ALFA=X(N+1)*ALFA/2.
      PRINT*, 'GOT BY B4'

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
AMBDA=1
IF (ALFA) 5,0,4
PRINT*,GOT BY B5
IF (ALFA-AMBDA) 5,0,6
PRINT*,GOT BY B6
AMBDA=ALFA
ALFA=0
DO 8 I=1,N
X(I)=X(I)+AMBDA*H(I)
PRINT*,GOT BY B7
CONTINUE
FX=FY
DX=DY
PRINT*,GOT BY B8
CALL FUNCT (N,X,F,G,RR)
PRINT*,GOT BY B9
IF (NFAIL.EQ.1) RETURN
PRINT*,GOT BY B10
NUMF=NUMF+1
IF (F.LT.FX) RETURN
PRINT*,GOT BY B11
FY=F
DY=0
DO 9 I=1,N
DY=DY+G(I)*H(I)
PRINT*,GOT BY B12
CONTINUE
PRINT*,GOT BY B13
IF (DY) 10,30,13
PRINT*,GOT BY B14
IF (FY-EX) 11,13,13
PRINT*,GOT BY B15
AMBDA=AMBDA+ALFA
ALFA=AMBDA
IF (N*AMBDA-1.E10) 7,7,12
PRINT*,GOT BY B16
IER=2
GO TO 31
PRINT*,GOT BY B17
T=0
IF (AMBDA) 15,30,15
PRINT*,GOT BY B18
Z=1+(EX-EX)/AMBDA+DX+DY
ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
DALFA=Z/ALFA
DALFA=DALFA+DALFA-DX/ALFA+DY/ALFA
IF (DALEA) 31,14,14
PRINT*,GOT BY B19
W=ALFA*SQR(DALFA)
ALFA=DY-DX+W
IF (ALFA) 17,18,17
PRINT*,GOT BY B20
ALFA=(DY-Z+W)/ALFA
GO TO 19
PRINT*,GOT BY B21
ALFA=(Z+DY+W)/(Z+DX+Z+DY)
ALFA=ALFA*AMBDA
DO 20 I=1,N
X(I)=X(I)+ALFA*H(I)
CONTINUE
CALL FUNCT (N,X,F,G,RR)
IF (NFAIL.EQ.1) RETURN
NUMF=NUMF+1
IF (F.LT.FX) GO TO 30
```



ORIGINAL PAGE IS  
OF POOR QUALITY

```

22 IF (F-FY) 30,30,22
23 DALFA=0
24 DO 33 I=1,N
25   GALEA=DALFA+G(I)*H(I)
26   CONTINUE
27 IF (DALFA) 24,27,27
28 IF (F-FX) 26,25,27
29 IF (DX-DALFA) 25,30,26
30 FX=F
31 DX=DALFA
32 T=ALFA
33 AMBDA=ALFA
34 GO TO 12
35 IF (F-F) 29,28,29
36 IF (DY-DALFA) 29,30,29
37 FY=F
38 DY=DALFA
39 AMBDA=AMBDA-ALFA
40 GO TO 13
41 RETURN
42 CONTINUE
43 IF (DY.GE.0.) IER=-2
44 IF (GNRM.LE.1.E-10) GO TO 32
45 IF (GNRM/GNRM.LE.EPS) IER=-3
46 CONTINUE
47 IF (DALFA.LT.0.) IER=-1
48 NFAIL=1
49 WRITE(5,33)
50 FORMAT(///,1X,' THE PROGRAM HAS FAILED')
51 RETURN
52
53 END
54 SUBROUTINE FUNCT (N,AL,U,GRAD,RR)
55 IMPLICIT REAL*8 (A-H,O-Z)
56
57 THIS SUBROUTINE IS USED TO CALCULATE THE OPTIMIZATION AND THE
58 GRADIENT AT ANY GIVEN POINT FOR SUBROUTINE POPT
59
60 DIMENSION AL(*), GRAD(*), SUM(17), RR(*)
61 COMMON /FAIL/ NFAIL
62 COMMON /HELP/ S(101), XX(16,101), C(8), M
63 C.... ABOVE LINE CHANGED FROM TEXT
64
65 N21=2*N+1
66 ZERO=0.0
67 DO 1 I=1,N21
68   SUM(I)=0.0
69   PRINT*, 'GOT BY C1'
70   CONTINUE
71   DO 4 I=1,M
72     SZ=ZERO
73     DO 3 K=1,N
74       SZ=SZ+AL(K)*XX(K,I)
75     PRINT*, 'GOT BY C2'
76     CONTINUE
77     IF (SZ.GT.74.) GO TO 9
78     PRINT*, 'GOT BY C3'
79     SS=EXP(SZ)*XS(I)
80     SUM(1)=SUM(1)+SS

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

GO 4 J=1,N1
SUM(J)=SUM(J)+XX(J-1,I)*SS
PRINT*,GOT BY C4,
CONTINUE
DO 5 I=1,N2
SUM(I)=SUM(I)+SUM(J)
PRINT*,GOT BY C5,
CONTINUE
U=0.0
DO 6 I=1,N
RR(I)=(SUM(I+1)-C(I))/C(I)
U=U+RR(I)*RR(I)
PRINT*,GOT BY C6,
CONTINUE
DO 8 N=1,N
GRAD(K)=0.0
DO 7 J=1,N
GRAD(K)=GRAD(K)+(SUM(J+1)*SUM(K+1))*RR(J)/C(J)
PRINT*,GOT BY C7,
CONTINUE
GRAD(K)=GRAD(K)*2,
PRINT*,GOT BY C8,
CONTINUE
PRINT*,GOT BY C9,
RETURN,GOT BY C10,
CONTINUE
AA=S2-32
ZERO=ZERO-AA
GO TO 2
PRINT*,GOT BY C11,
END
55
SUBROUTINE START (XMAX,XMIN,ALAMDA,KSTART,CC,NL,IPRINT,UMIN,MODE,M
1AXFN,ETA)
IMPLICIT REAL*8 (A-H,O-Z)
THIS SUBROUTINE IS USED TO FIND A REASONABLE STARTING POINT FOR
SUBROUTINE MPOPT
DIMENSION R(11), ETA(*)
DIMENSION CC(X), X(10), Y(10), W(10,10)
DIMENSION ALAMDA(X), X(10), Y(10), W(10,10)
COMMON/HELP/S(101),XX(10,10),C(8),M
ABOVE LINE CHANGED FROM TEXT
COMMON /FAIL/ NFAIL
GO TO (3,1,5,26), KSTART
CONTINUE
NEALL=0
DO 2 I=1,NL
ALAMDA(I)=0.0
CONTINUE
RETURN
CONTINUE
NFAIL=0
ALAMDA(1)=CC(1)/CC(2)
ALAMDA(2)=S/CC(2)
DO 4 I=1,NL
ALAMDA(I)=0.0
CONTINUE
RETURN
CONTINUE

```

04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45

```

NNN=NL/2
NNN=NNN*2
NP1=NL+1
DELTA=(XMAX-XMIN)/FLOAT(NL)
DO 6 I=1,NP1
X(I)=XMIN+FLOAT(I-1)*DELTA
CONTINUE
IF (NNN.NE.NL) GO TO 19
W(1,1)=1
W(1,NP1)=1
DO 7 I=2,NL,2
W(1,I)=4.
CONTINUE
IF (NL.EQ.2) GO TO 9
NM1=NL-1
DO 8 I=3,NM1,2
W(1,I)=3.
CONTINUE
CONTINUE
DO 10 J=1,NP1
DO 10 I=2,NP1
W(I,J)=W(I-1,J)*X(J)
Y(1)=3./DELTA
DO 11 I=1,NL
Y(I+1)=C(I)*Y(1)
CONTINUE
CALL SOLVE (W,Y,XID,NP1,10)
CONTINUE
DO 13 I=1,NP1
DO 13 J=1,NP1
W(I,J)=0
U6
DO 14 I=1,NP1
IF (Y(I).LE.0.0) Y(I)=.0002
CONTINUE
DO 15 I=1,NP1
Y(I)=ALOG(Y(I))
CONTINUE
DO 16 I=1,NP1
W(I,1)=1.
CONTINUE
DO 17 I=2,NP1
DO 17 J=1,NP1
W(J,I)=W(J,I-1)*X(J)
CALL SOLVE (W,Y,XID,NP1,10)
DO 18 I=1,NL
ALAMDA(I)=Y(I+1)
RETURN
CONTINUE
R(1)=3./8.
R(4)=3./8.
R(2)=9./8.
R(3)=9./8.
IF (NL.EQ.3) GO TO 22
R(NL+1)=1./3.
R(4)=R(4)+1./3.
DO 20 I=5,NL,2
R(I)=2./3.
CONTINUE
IF (NL.EQ.5) GO TO 22
NS=NL-1
DO 21 I=6,NS,2
R(I)=2./3.

```

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

```

21 CONTINUE
22 DO 23 I=1,NP1
   W(1,I)=R(I)
23 CONTINUE
24 DO 24 J=1,NP1
   W(1,J)=W(1-I,J)*X(J)
   Y(1)=1/DELTA
   DO 25 I=1,NL
     Y(I)=5(I)*Y(I)
25 CONTINUE
   CALL SOLVE (W,Y,XID,NP1,10)
   GO TO 12
26 CONTINUE
   N=2
   ALAMDA(2)=-.5/CC(2)
   ALAMDA(1)=CC(1)/CC(2)
   N=1
27 CONTINUE
   ALAMDA(N+1)=2.0
   ALAMDA(N+2)=0.0
   PRINT*,GOT BY A'
   CALL MPOPT (ALAMDA,N,STA,UMIN,MAXFN,MODE,IPRINT)
   PRINT*,GOT BY B'
   IF (NFAIL.EQ.1) RETURN
   IF (N.EQ.NL) RETURN
   ALAMDA(N+1)=0.0
   N=N+1
   GO TO 27
END

51 SUBROUTINE SOLVE (A,X,XID,N,NA)
   IMPLICIT REAL8 (A-H,O-Z)
   DIMENSION A(NA,*), X(*)
   D=0
   DATA DIV/.693147181/
   DO 1 I=1,N
     AA=0
     DO 1 J=I,N
       AB=ABS(A(J,I))
       IF (AB+LE+AA) GO TO 1
       K=J
     END DO
     AA=AB
     CONTINUE
     D=DSLOG(AA)
     IF (I.EQ.N) GO TO 2
     IF (K.EQ.I) GO TO 3
     DO 2 J=I,N
       AB=A(J,I)
       A(I,J)=A(K,J)
       A(K,J)=AB
     CONTINUE
     X(I)=X(K)
     X(K)=AB
     I=I+1
     DO 3 J=I,N
       AA=A(J,I)/A(I,I)
       A(J,I)=0
       DO 4 K=I+1,N
         A(J,K)=A(J,K)+AA*A(I,K)
       CONTINUE

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
5      CONTINUE
6      CONTINUE
7      XID=DOU
8      X(N)=X(N)/A(N,N)
9      DO 9 I=2,N
10     I=N+1-I
11     I1=I+1
12     AA=0.
13     DO 9 J=I1,N
14     AA=AA+(I1-J)*X(I1-J)
15     CONTINUE
16     X(I)=(X(I)-AA)/A(I,I)
17     CONTINUE
18     RETURN
19     END

20     SUBROUTINE SIMSON
21     IMPLICIT REAL*8 (A-H,O-Z)
22
23     C THIS SUBROUTINE IS TO CALCULATE THE SIMPSON MULTIPLIERS
24
25     C.... COMMON/HELP/S(101),XX(16,101),C(8),M
26     C      ABOVE LINE CHANGED FROM TEXT
27     S(1)=1.
28     S(M)=1.
29     N=M-1
30     DO 1 I=2,N,2
31     S(I)=4.
32     CONTINUE
33     N=N-1
34     DO 2 I=3,N,2
35     S(I)=2.
36     CONTINUE
37     RETURN
38     END

39     SUBROUTINE MULT (XMAX,XMIN,N)
40     IMPLICIT REAL*8 (A-H,O-Z)
41     FUNCTION
42
43     C.... COMMON/HELP/S(101),XX(16,101),C(8),M
44     C      ABOVE LINE CHANGED FROM TEXT
45     DELTA=(XMAX-XMIN)/FLOAT(N-1)
46     DO 1 I=1,M
47     XX(1,I)=XMIN+FLOAT(I-1)*DELTA
48     NN=2*N
49     DO 1 J=2,NN
50     XX(J,I)=XX(J-1,I)*XX(1,I)
51     CONTINUE
52     RETURN
53     END

54     SUBROUTINE CONVER (CM,NL)
55     IMPLICIT REAL*8 (A-H,O-Z)
56
57     C THIS SUBROUTINE IS TO CALCULATE THE MOMENTS ABOUT THE ORIGIN
58     DIMENSION CM(*)
```

```

COMMON=HELP-S+0+--+---++-+0+-e'9)-H
C... ABOVE LINE CHANGED FROM TEXT
      C(1)=CM(I)
      IF (NL.EQ.1) RETURN
      DO 2 J=2,NL
        CJ=C(J)-C(1)**J*(-1.)**J
        NJ=-J-1,N
        DO 1 K=1,N
          CC(C(J))*FACTO(J)/(FACTO(K)*FACTO(J-K))*C(1)**(K)*C(J-K)
          CONTINUE
        CONTINUE
      END

SUBROUTINE TRN1 (X1MAX,X1MIN,C,X2MAX,X2MIN,NL)
IMPLICIT REAL*8 (A-H,O-Z)

THIS SUBROUTINE IS USED TO CALCULATE THE MOMENTS FOR THE MODIFIED
LIMITS

DIMENSION C(1)
SCL=(X1MAX-X1MIN)/2*(X2MAX-X2MIN)
C(1)=C(1)/SCL-X1MIN/SCL*X2MIN
IF (NL.EQ.1) RETURN
DO 1 I=2,NL
  C(I)=C(I)/SCL**(FLOAT(I))
CONTINUE
RETURN
END

SUBROUTINE TRN2(X1MAX,X1MIN,X,X2MAX,X2MIN,N)
IMPLICIT REAL*8 (A-H,O-Z)
THIS SUBROUTINE IS AN ALTERNATIVE TO TRN2 (BELOW)

..... CALCULATES THE LAGRANGIAN MULTIPLIERS FOR A DIFFERENT INTERVAL
..... DOUBLE PRECISION VERSION
..... DOUBLE PRECISION S,A,B,DZ(10),FAC,DX1MAX,DX1MIN,DX2MAX,DX2MIN
..... DIMENSION X(*)
..... DX1MAX=X1MAX
..... DX1MIN=X1MIN
..... DX2MAX=X2MAX
..... DX2MIN=X2MIN
      NP1=N+1
      DO 10 I=1,NP1
        DX(I)=X(I)
        S=(DX1MAX-DX1MIN)/(DX2MAX-DX2MIN)
        A=DX2MIN-DX1MIN/S
        DX(I)=DX(1)-ALOG(S)
        DO 1 I=1,N
          DX(I)=DX(1)+DX(I+1)*A**I
          CONTINUE
        IF (NEQ.1) GO TO 6
        DO 3 J=2,N
          DO 3 I=J,N
            FAC=1.
            KK=I-J+2
            DO 2 K=KK,I
              FAC=FAC*DBLE(FLOAT(K))
            CONTINUE
          DX(J)=DX(J)+FAC/DBLE(FACTO(J-1))*A**((I-J+1)*DX(I+1))
          CONTINUE
        DX(J)=DX(J)/S**((J-1)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

CONTINUE
CONTINUE
DO 11 I=1,NPI
  X(I)=OX(I)
RETURN
END

```

```

SUBROUTINE TRN2 (X1MAX,X1MIN,X,X2MAX,X2MIN,N)
IMPLICIT REAL*8 (A-H,O-Z)

```

THIS SUBROUTINE IS USED TO CALCULATE THE LAGRANGIAN MULTIPLIERS  
AT THE ORIGINAL LIMITS

```

DIMENSION X(1)
S=(X1MAX-X1MIN)/(X2MAX-X2MIN)
A=X1MIN-X1MIN/S
X(1)=X(1)-ALOG(S)

```

```

DO 1 I=1,N
  X(I)=X(1)+X(I+1)*A**I
CONTINUE
IF (N.EQ.1) GO TO 5
DO 3 J=2,N
  DO 2 I=J,N
    FAC=1
    DO 2 K=KK,I
      FAC=FAC*FLOAT(K)
CONTINUE
X(J)=X(J)+FAC/FACTO(J-1)*A**((I-J+1)*X(I+1))
CONTINUE
X(J)=X(J)/S**((J-1))
CONTINUE
X(N+1)=X(N+1)/S**N
RETURN
END

```

```

DO 2 K=KK,I
  FAC=FAC*FLOAT(K)
CONTINUE
X(J)=X(J)+FAC/FACTO(J-1)*A**((I-J+1)*X(I+1))
CONTINUE
X(J)=X(J)/S**((J-1))
CONTINUE
X(N+1)=X(N+1)/S**N
RETURN
END

```

60

```

FUNCTION CDF (XMIN,XMAX,XP,AL,N)
IMPLICIT REAL*8 (A-H,O-Z)

```

THIS FUNCTION SUBROUTINE IS TO CALCULATE THE CUMULATIVE-DISTRIBUTION FUNCTION AT A GIVEN POINT

```

INPUT
  XMIN = LOWER BOUND
  XMAX = UPPER BOUND
  XP = SPECIFIED POINT
  AL(I) = ARRAY OF PARAMETERS, DIMENSION N
  N = NUMBER OF PARAMETERS

```

```

DIMENSION AL(1)
IF (XP.LE.XMIN) GO TO 3
IF (XP.GE.XMAX) GO TO 4
RANGEX=XMAX-XMIN
RANGEX=XP-XMIN
SS=RANGEX/RANGEX*51.
JSS=SS
JSS=(JSS/2)*2+5
AREA=0.0
JSM1=JSS-1
DELTA=RANGEX/FLOAT(JSM1)
DO 1 I=2,JSM1,2
  X=XMIN+FLOAT(I-1)*DELTA
  AREA=AREA+4.*ENTRPF(AL,N,X)

```

```

JSM1=JSS-1
DELTA=RANGEX/FLOAT(JSM1)
DO 1 I=2,JSM1,2
  X=XMIN+FLOAT(I-1)*DELTA
  AREA=AREA+4.*ENTRPF(AL,N,X)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

1  CONTINUE
   JSM1=JSM1-I
   DO 2 I=3,JSM1/2
     X=XMIN+FLOAT(I-1)*DELTA
     AREA=AREA+2.*ENTRPF(AL,N,X)
   CONTINUE
   AREA=AREA+ENTRPF(AL,N,XMIN)+ENTRPF(AL,N,XP)
   AREA=AREA*DELTA/3.
   CDF=AREA
   GO TO 3
2  CDF=0.0
   GO TO 5
3  CDF=1.
   CONTINUE
   RETURN
   END

FUNCTION ENTRPF (AL,NPL,X)
  IMPLICIT REAL*8 (A-H,O-Z)
  FUNCTION TO EVALUATE THE ENTROPY DENSITY FUNCTION AT A GIVEN POINT
  INPUT
    AL(I) = ARRAY CONTAINING PARAMETERS, DIMENSION NPL
    NPL = NUMBER OF PARAMETERS
    X = GIVEN VALUE
  DIMENSION AL(*)
  S=AL(1)
  DO 1 I=2,NPL
    S=S+AL(I)*X***(I-1)
  CONTINUE
  ENTRPF=EXP(S)
  RETURN
  END

C  FUNCTION FACTO (M)
C  IMPLICIT REAL*8 (A-H,O-Z)
C.... CALCULATES FACTORIAL OF M
  FACTO=1
  IF (M.EQ.0) RETURN
  DO 1 I=1,M
    FACTO=FACTO*FLOAT(I)
  CONTINUE
  RETURN
  END

```



[illegible]

62

```
EEEEEEEEE  
AAAAAAA  
Digital Equipment Corporation - VAX/VMS Version U4.7  
AAAAAAA
```

ORIGINAL PAGE IS  
OF POOR QUALITY

300.0000	45.0000
3.0000	0.3000
500.0000	75.0000
7.0000	0.7000
250.0000	12.5000
20.0000	1.0000
150.0000	7.5000
0.5000	0.0150
1500.0000	75.0000
20.0000	2.0000
850.0000	25.0000

Digital Equipment Corporation - VAX/VMS version 4.4-1  
Digital Equipment Corporation - VAX/VMS version 4.4-1

[illegible][illegible]

File DBAO:CRANDM44.CPR;1 (383,942,0), last revised on 23-NOV-1988 11:21, is a 51 block sequential file owned by UIC [11,11]. The records are variable length with FORTRAN (FTN) carriage control. The longest record is 132 bytes.

Job RANDM44 (685) queued to SVS8SBPRI on 23-NOV-1988 11:22 by user NETNONPRI4. UIC [11,11] under account 201COADD at priority 100. Started on printer JPF7 on 23-NOV-1988 11:22 from queue JPF7.

[illegible]

ORIGINAL PAGE IS  
OF POOR QUALITY

[illegible]

[illegible]

[illegible]

# INPUT DATA FOR SUBROUTINE MEP:

INPUT DATA IS PRINTED OUT FOR KDATA = 1 ONLY . . . KDATA =  
 INTERMEDIATE OUTPUT EVERY KPRINT(TH) CYCLE . . . KPRINT =  
 NUMBER OF KNOWN FIRST MOMENTS . . . N =  
 HIGHER LIMIT . . . KMAX =  
 LOWER LIMIT . . . KMIN =  
 FIRST MOMENTS . . . CC(1) =  
 THE ALLOWED TOLERANCE IN LAGRANGIAN EQUATIONS . . . TOL =  
 THE CUMULATIVE DISTRIBUTION REQUIRED AT NXP POINTS. NXP =

0.370334345E+00 0.178168955E+00 0.752378156E+00

ORIGINAL PAGE IS  
 OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

INTERMEDIATE RESULTS FOR SUBROUTINE MEP

NUMBER OF INTEGRATION STATION M = 31  
MODIFIED MAXIMUM AND MINIMUM LIMITS X2MAX = 0.100000000E+00 X2MIN = 0.000000000E+00  
MODIFIED MOMENTS ABOUT THE EXPECTED VALUE CC(1) = 0.416759124E+00 CC(2) = 0.372395000E-01  
MODIFIED MOMENTS ABOUT THE ORIGIN C(1) = 0.416759124E+00 C(2) = 0.210912118E+00 C(3) = 0.121897179E-00 C(4) = 0.771597012E-01  
SUBROUTINE MPROT TOLERANCES ETA(1) = 0.100000000E-11 ETA(2) = 0.100000000E-05 ETA(3) = 0.100000000E-22 ETA(4) = 0.100000000E-20

NORMAL ASSUMPTION STARTING METHOD

STARTING VALUES

AL(1) = 0.111959940E+02 AL(2) = 0.134322122E+02 AL(3) = 0.109000000E+00 AL(4) = 0.100000000E+00

CYC NO.	NUMF	NORMGRAD RESIDUALS	TOTAL X(1)	X(2)	VARIABLES X(3)	X(4)	R(1)	R(2)	RESIDUALS R(3)	R(4)	
0	2	0.17606E-01	0.23719E-02	0.11222E+02	-0.13405E-02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
1	4	0.32974E-02	0.32044E-02	0.12622E+02	-0.13405E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
2	5	0.51891E-02	0.52080E-04	0.80751E+01	-0.10197E+01	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
3	7	0.19415E-02	0.25700E-04	0.77949E+01	-0.05481E+01	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
4	8	0.42977E-02	0.31381E-04	0.21444E+02	-0.42911E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
5	10	0.27787E-02	0.23060E-04	0.17639E+02	-0.47370E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
6	11	0.24290E-02	0.18863E-04	0.20061E+02	-0.56621E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
7	13	0.10508E-02	0.10587E-04	0.23892E+02	-0.67284E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
8	15	0.13074E-02	0.10539E-04	0.23238E+02	-0.68597E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
9	16	0.35514E-03	0.36431E-03	0.26844E+02	-0.79546E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
10	19	0.12110E-03	0.36089E-03	0.27017E+02	-0.80387E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
11	20	0.27919E-03	0.25729E-03	0.26117E+02	-0.77333E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
12	21	0.34299E-03	0.12052E-03	0.27481E+02	-0.81242E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
13	23	0.14400E-03	0.72443E-06	0.28685E+02	-0.85797E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
14	25	0.14845E-03	0.71197E-06	0.28912E+02	-0.86679E+02	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
15	27	0.43064E-03	0.47545E-06	0.36093E+02	-0.11405E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
16	29	0.34003E-03	0.40703E-06	0.34328E+02	-0.10818E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
17	31	0.63374E-04	0.38902E-06	0.34313E+02	-0.10719E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
18	32	0.14003E-03	0.30074E-06	0.35067E+02	-0.11034E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
19	34	0.59595E-04	0.26184E-06	0.34631E+02	-0.10878E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
20	37	0.91125E-04	0.26054E-06	0.35094E+02	-0.11034E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
21	39	0.21647E-03	0.20172E-06	0.35750E+02	-0.11341E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
22	41	0.24074E-03	0.20095E-06	0.35869E+02	-0.11388E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
23	43	0.18373E-03	0.19373E-06	0.36052E+02	-0.11436E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
24	43	0.13020E-03	0.17917E-06	0.36163E+02	-0.11496E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01
25	44	0.19623E-03	0.13847E-06	0.37770E+02	-0.12126E+03	0.23839E-01	0.19798E-01	0.224E-01	0.210E-01	0.41E-02	0.375E-01







Digital Equipment Corporation - VAX/VMS Version V4 -

[illegible]

File DBAO: [JPL01.CPR.1 (359,204,0)], last revised on 23-NOV-1988 11:20, is a 2 block sequential file owned by UIC [11,11]. The records are variable length with FORTRAN (FTN) carriage control. The longest record is 25 bytes.

Job JPL01 (683) queued to SYSSEBRT on 23-NOV-1988 11:20 by user NETNOVPRV, UIC [11,11], under account 20100ADD at priority 100. started on printer \_IT7\_ on 23-NOV-1988 11:20 from queue IT7.

[illegible]

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

[illegible]



ORIGINAL PAGE IS  
OF POOR QUALITY

(E12. 4. 1X. E12. 4)  
0. 5723E+01 0. 0000E+00  
0. 5919E+01 0. 3801E-02  
0. 6115E+01 0. 1878E-01  
0. 6311E+01 0. 5763E-01  
0. 6506E+01 0. 1267E+00  
0. 6702E+01 0. 2208E+00  
0. 6898E+01 0. 3237E+00  
0. 7094E+01 0. 4288E+00  
0. 7289E+01 0. 5217E+00  
0. 7485E+01 0. 6038E+00  
0. 7681E+01 0. 6758E+00  
0. 7876E+01 0. 7378E+00  
0. 8072E+01 0. 7981E+00  
0. 8268E+01 0. 8517E+00  
0. 8464E+01 0. 8972E+00  
0. 8659E+01 0. 9371E+00  
0. 8855E+01 0. 9687E+00  
0. 9051E+01 0. 9874E+00  
0. 9246E+01 0. 9953E+00  
0. 9442E+01 0. 9973E+00  
0. 9638E+01 0. 1000E+01

## 9.0 APPENDIX D

### IMSL SUBROUTINE CALLS FROM RANDOM3 AND RANDOM4

#### RANDOM3

1. RNSET - Initializes a random seed for use in the IMSL random number generators.
2. RNNOR - Generates pseudorandom numbers from a standard normal distribution using an inverse CDF method.
3. RNLNL - Generates pseudorandom numbers from a lognormal distribution.
4. DESPL - Performs nonparametric probability density function estimation by the penalized likelihood method.
5. GCDF - Evaluates a general continuous cumulative distribution function given the ordinates of the density.

#### RANDOM4

1. RNSET - Initializes a random seed for use in the IMSL random number generators.
2. RNNOR - Generates pseudorandom numbers from a standard normal distribution using an inverse CDF method.
3. RNLNL - Generates pseudorandom numbers from a lognormal distribution.

## 10.0 APPENDIX E

### SAMPLE SAS/GRAPH PROGRAM FOR RANDOM3 AND RANDOM4

```
data a;
INFILE 'PLOT1.CPR' FIRSTOBS=2;input x y;
GOPTIONS DEVICE=HP7470;
proc gplot;
  axis1 label=(h=1 f=simplex 'LOG OF CYCLES')
        value=(h=1 f=simplex);
  axis2 value=(h=1 f=simplex) label=none;
  plot y*x / haxis=axis1 vaxis=axis2;
  TITLE H=1 A=90 F=SIMPLEX 'PROBABILITY DENSITY FUNCTION';
  symbol i=spline v=square;
data B;
INFILE 'PLOT2.CPR' FIRSTOBS=2;input x y;
proc gplot;
  axis1 label=(h=1 f=simplex 'LOG OF CYCLES')
        value=(h=1 f=simplex);
  axis2 value=(h=1 f=simplex) label=none;
  plot y*x / haxis=axis1 vaxis=axis2;
  TITLE H=1 A=90 F=SIMPLEX 'CUMULATIVE DISTRIBUTION FUNCTION';
  symbol i=spline v=square;
```